FUEL ESTIMATION USING
DYNAMIC RESPONSE

THESIS

Michael S. Hines, Second Lieutenant, USAF

AFIT/GA/ENY/07-M12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/ENY/07-M12

FUEL ESTIMATION USING

DYNAMIC RESPONSE

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

Michael S. Hines, B.S.

Second Lieutenant, USAF

March, 2007

AFIT/GA/ENY/07-M12

FUEL ESTIMATION USING
DYNAMIC RESPONSE

Michael S. Hines, B.S.

Second Lieutenant, USAF

Approved:

/signed/	22 Mar 2007
LtCol Nathan A. Titus, Ph.D. (Chairman)	date
/signed/	22 Mar 2007
Dr Richard G. Cobb (Member)	date
/signed/	22 Mar 2007
Maj Eric D. Swenson, Ph.D. (Member)	date

AFIT/GA/ENY/07-M12

Abstract

New regulations governing satellites in geostationary orbit require satellites to transfer into a parking orbit as part of the decommissioning process. These regulations increase the demand for accurate fuel estimation techniques for satellites. This study estimates the change in fuel mass from the dynamic response of the Air Force Institute of Technology's simulated satellite (SimSAT) to known control inputs. With an iterative process, the moment of inertia of SimSAT about the yaw axis was estimated by matching a model of SimSAT to the measured angular rates. A change in fuel mass was then estimated from the known relation between the change in moment of inertia to the change in fuel mass.

Fuel masses of 1, 2, and 3 kilograms were estimated. The fuel estimation process developed in this study was able to estimate the fuel as $1.5664 \pm 3.7157\ kg$, $2.8880 \pm 3.8875\ kg$, and $3.9114 \pm 3.4648\ kg$ respectively. While the theory behind the estimation process is sound, the implementation still requires work.

Acknowledgements

I thank Lt. Col. Titus for all of his advice. I also acknowledge my committee members, Dr. Cobb and Maj. Swenson for their insight. Their help gave me the guidance I needed to make a better end product.

I would like to thank Mr. Wilber Lacy for all of the help he gave me getting SimSAT up and running. Without his help I would have never been able to finish. I would also like to thank Mr. Jay Anderson for his creativity creating the tanks to attach to SimSAT. Most of all, I thank my loving family and my wonderful girlfriend. Their love and support helped me make it through AFIT.

Michael S. Hines

Table of Contents

List of Figures

List of Tables

<center>*List of Symbols*</center>

Symbol   Page

List of Abbreviations

Abbreviation		Page
Abbreviation		Page

FUEL ESTIMATION USING

DYNAMIC RESPONSE

I. Introduction

One of the most prized resources for satellites in space is fuel. The current limitations of technology make the task of refuelling satellites in orbit difficult and not cost effective. Fuel is required for orbit keeping maneuvers and in some cases attitude control, making it mission essential to almost all satellites. Without the ability to refuel satellites, the amount of fuel becomes a major factor in calculating a satellite's projected lifespan. As spacecraft continue to remain operational beyond their projected lifespan, mission planners and operators have become increasingly interested in determining precisely how much fuel remains in order to extend the life of the satellite as much as possible. In addition, the changing political environment dealing with space use provides additional motivation to quantify the remaining fuel.

1.1 Motivation

Geostationary (GEO) and Low Earth Orbits (LEO) are the most commercially used and therefore highly valuable. In the past, satellites were used until the fuel ran out or some other issue caused a satellite to become useless, leaving an inoperable satellite orbiting Earth as space debris. The increase of space debris has caused the probability of collisions resulting in mission failure of operating satellites to increase. This problem has been an ongoing development for companies and countries with space assets. A United States interagency report published in November of 1995 discussed the build up of space debris, the current measures being taken to counter the problem, and recommendations for future action [5]. This report was followed by a technical report from the United Nations in 1999, which restated the seriousness of the problem [7]. After the United Nations report, the Inter-Agency Space Debris Coordination Committee (IADC) published general recommendations in 2002 to combat the increasing problem of space debris. One of many

suggestions was to create a protected area around GEO where only active satellites could remain [15]. Many of the suggestions made by IADC have been partially implemented over the years by individual satellite owners, but the United States was the first to implement them in official policy. The Federal Communication Commission (FCC) released new rules and regulations in June of 2004 implementing several of the past recommendations [6].

Of the new rules regulating spacecraft licensed by the FCC to provide services to the United States, only the rules applying to post-mission disposal of Satellites in GEO are relevant to this thesis. The new FCC rules require spacecraft in GEO to move into a disposal orbit with a perigee greater than or equal to the value calculated by the IADC formula in Equation 1.1.

$$36,021 + (1000 * C_R * A/m) \qquad (1.1)$$

C_R is the solar radiation pressure coefficient, A is the cross sectional area, and m is the total mass of the spacecraft without any fuel. Satellite owners are required to transfer their satellites in GEO into disposal orbits at the satellite's respective end-of-life (EOL) point. Moving satellites away from GEO protects future and current GEO satellites from possible collisions and explosions. In the past, EOL occurred when the satellite had no remaining fuel and subsequently became uncontrollable and unreliable. With the new FCC rules, the EOL will occur when the amount of fuel remaining is equivalent to the amount of fuel required for the final maneuver into the prescribed disposal orbit. The increased demand on fuel will cause a reduction in every GEO satellite's lifespan, which has a direct correlation to the revenue collected by the owners. The change in the EOL will decrease the operating time by one to three months, which equates to a loss of hundreds of billions of dollars throughout the satellite community [6].

Before the new FCC regulations were implemented, the amount of fuel remaining on a satellite was only monitored in order to decide the time frame in which a replacement satellite would need to be launched to insure uninterrupted service. Now, remaining fuel has become a critical factor in accurately determining when the final disposal maneuver needs to occur. The increased desire to know the remaining fuel has increased the scrutiny of the processes used to estimate a satellite's remaining fuel.

1.2   Fuel Estimation

The fuel estimation process has received increased attention since knowing the amount of remaining fuel is required to properly time a satellite's transfer to the designated disposal orbit. The EOL for many of the current and all of the future GEO satellites with FCC licenses is presently more closely related to fuel estimation than ever before. Fuel estimation is a common process, but the fine points of the process go unnoticed to the general public.

Estimation is a statistical operation that provides an estimated value that has the highest probability of being the true value. In addition, estimation provides a measure of confidence in the estimated value through what is called the standard deviation. A visual definition of standard deviation for a Gaussian distribution can be seen in Figure 1.1. A Gaussian distribution is assumed because a large number of measurements from properly calibrated equipment generates a Gaussian distribution about the true value being measured. The red, or middle section, highlights the range of measurements that are one standard deviation away from the true value, and include approximately 68% of all measurements made. The middle and adjacent sections highlight two standard deviations and include approximately 95% of all the measurements. Both the estimated value and the standard deviation are important to any estimation process, as they carry the majority of the beneficial information from the original data.
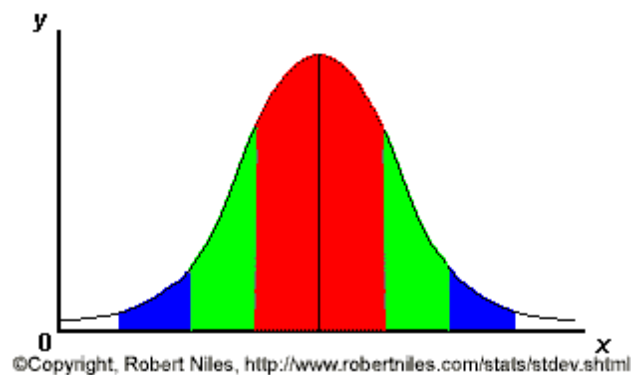
Figure 1.1   Normal Distribution with Standard Deviation [4]

There are a number of ways to estimate fuel, and it is the accuracy of each method that sets them apart. A method that provides an estimate with a higher confidence level (smaller standard deviation) allows the satellite to remain in orbit longer. This longer lifespan allows more profits to be earned from the large investment required to put a satellite into orbit. When the amount of fuel is estimated by a method that has a poor confidence level, the satellite owner must err on the side of caution and assume that the lowest reasonably possible estimate is the actual amount of remaining fuel. If a method provides an estimation with a large standard deviation, then the satellite's owner is required to reposition the satellite into its disposal orbit earlier than necessary. By decommissioning a satellite early, the satellite will have extra fuel once it reaches the parking orbit that could have been used to prolong the satellite's lifespan in GEO. In other words, the owner has unnecessarily reduced the satellite's lifespan beyond the affect of the new FCC regulations. In addition, the owner has lost the revenues the satellite would have earned over that time along with the capabilities provided by the satellite.

1.3   Current Fuel Estimation Methods

The high accuracy pressure transducer (HAPT) has an accuracy of $\pm 0.01\%$ full scale at the Beginning-of-Life (BOL) and an accuracy of $\pm 0.05\%$ full scale at the EOL. This is a large step up from the $\pm 0.5\%$ full scale accuracy of the average sensor currently in use. In addition, the Ultrasonic Flow Meter (UFM) was developed with an accuracy of $\pm 0.05\%$ full scale during liquid apogee engine firing, while typical meters currently in use have accuracies between $\pm 0.6\%$ to $\pm 2\%$ full scale [17]. Each of these new devices is designed to improve upon a different method of estimating fuel.

One method is the "ρVT method" that implements the ideal gas law with pressure and temperature as inputs to calculate the remaining fuel. Another method is the "gas injection method" that uses a rise in pressure and temperature from adding a known amount of gas into the fuel tanks. The measurements that both of these methods require are provided by the HAPT device at increased accuracies. The new measurements enable the estimation method to have higher confidence levels, thus allowing the satellite's lifespan to be extended. A third method is the "pulse count" or bookkeeping method. Correlating

ground calibration tests with telemetry data from the satellite of the thruster firings allows the used fuel to be estimated and tracked. The UFM device will drastically increase the confidence level of the estimates from the bookkeeping method [17]. Another method used on the ASTRIUM-SAS communication satellites is the "Thermal Propellant Gauging Technique" (TPGT). This technique characterizes the propagation of thermal energy through the fuel tank in order to estimate the amount of fuel remaining. Specifically, the TPGT method estimates the time constant that describes the thermal inertia of the fuel tank [9].

The best approach for fuel estimation is to implement as many different methods that are as accurate as possible and then to combine their estimations using a least squares algorithm. Integrating several methods results in an estimation with a higher confidence level than a single method would provide. An important note to remember when using a least squares algorithm is that the individual methods should be as independent as possible [9]. The measured quantities along with the instruments used by each method should be different in order to ensure that the results from the various methods can be assumed to be independent of one another. For example, the "ρVT method" and the "gas injection method" use the same measurements from the same instruments, just under slightly different situations. Therefore, combining the results from these techniques would not offer a significant improvement in accuracy over either method used alone. On the other hand, either of the two methods could be combined with the bookkeeping method because the measurements and instruments used in the previous two methods are completely independent of those used in the bookkeeping method. The objective of this thesis is to provide a third method that is independent of the methods discussed here in order to provide an additional method to integrate into the least squares algorithm.

1.4 Research Objectives

The Air Force Institute of Technology's (AFIT) Simulated SATellite (SimSAT) has almost complete three axis freedom of movement. The characteristics of SimSAT are discussed in detail in Chapter III. The objective of this thesis is to demonstrate a method of estimating the remaining fuel with the SimSAT hardware. This is achieved by determining

the change in the moment of inertia (MOI) found using the dynamic response of SimSAT from known attitude actuator commands. The effectiveness of the method was investigated using fuel tanks attached to the the SimSAT hardware. The following objectives are the goals set for this thesis:

1. Fix 'bugs' within SimSAT,

   - Characterize reaction wheels

   - Fix gyro data corruption problem

   - Get main SimSAT program working correctly

   - Streamline SimSAT code

2. Create an experimental setup to test the fuel estimation process,

3. Create an algorithm to calculate the remaining fuel from experimental data,

4. Make several estimations of different fuel masses in order to characterize the fuel estimation process.

1.5   Thesis Outline

Chapter I introduced the motivation for this thesis while Chapter II covers the background and theory supporting the process used to meet the thesis objectives. Chapter III discusses the specific equipment used to achieve the main objectives along with the preparatory work required to get the equipment working properly. Finally, Chapter IV discusses the results, and Chapter V discusses the conclusions.

II. Background

2.1   Literary Review

2.1.1   System Identification.   On-orbit system identification has been an ongoing development since many satellites have become too large and sensitive to characterize them fully in the gravity environment and the atmospheric conditions found on the surface of the Earth [8]. Using a set of known inputs and measurements of corresponding reactions we can create state space or transfer function models that accurately describe the system or satellite. Even smaller satellites that can be fully characterized on Earth benefit from on-orbit system identification due to the changes in the system from the large vibrational forces experienced during launch. There are the more traditional methods such as the least squares algorithm or a relatively newer group of methods referred to as "Subspace-based State-Space Systems IDentification (4SID)" methods that are designed to create accurate mathematical models capable of describing a satellite's behavior under known conditions [24].

A common use of on-orbit system identification is characterizing a satellite's orbit. When satellites were small and relatively simple, the satellites' characteristics were able to be characterized prior to launch. The increase in the complexity and size of satellites has created the need to characterize a satellite's mass and configuration properties in orbit. The same techniques applied to the characterization of a satellite's orbit can also be applied to the attitude and therefore to the mass properties of the satellite. Satellite attitude is controlled by the onboard computer with little input from the operators. Using devices, such as star trackers and gyros, the onboard computer can maintain a specific attitude using feedback control with a predetermined control law that does not require complete knowledge of the satellite's characteristics and their affect on the satellite's dynamics.

The article by Shuichi Adachi and his co-authors discusses several techniques for on-orbit characterization of satellites, and applies them to the Engineering Test Satellite-VI. The techniques used include polynomial black-box methods and 4SID methods. The models that were developed were able to accurately describe the satellite's motion in both the frequency and the time domains [8]. The results from this article proves that it is possible

to accurately model a satellite's attitude dynamics, which is an important component of the fuel estimation technique developed in this thesis. The techniques used in the article could be modified in order to work with the fuel estimation technique developed in this thesis, but due to the limited time and the unfamiliarity with the methods they were not adapted for use in this thesis.

2.1.2 Modelling Fuel Slosh. Researchers at the Southwest Research Institute have developed a platform to test fuel slosh for spinning spacecraft [13, 19, 20]. Using an eight inch diameter spherical tank the accuracy of what was labelled as "Dodge" or "SLOSH" code was verified. The code provides parameters that allow the sloshing of a specific fluid in a tank to be estimated by a fixed mass and a surface wave described by a pendulum with a point mass on the end. The verification was done by experimentally matching the dynamics of sloshing water to that of a mechanical pendulum and a fixed mass. The results showed that the mechanical pendulum was able to model the dynamics of the sloshing water with a fair amount of accuracy. In addition, a figure from the article described the relationship between the percent fill and total mass, fixed mass, and the pendulum point mass or the sloshing mass (Figure 2.1). At lower fills the sloshing mass that is incorporated as a surface wave is dominant, and around the 60% fill level the stationary and sloshing masses are approximately equal. Beyond the 60% fill level the stationary mass is dominant. The sloshing mass peaks at 60%, so that is the fill level that would experience the largest affect from the sloshing fuel. In the same article, there was also a table describing the pendulum's length based on the percent fill seen in Table 2.1 [20]. As the fill level decreases the pendulum length increases.

Table 2.1   Pendulum Geometry for 8" Sphere [20]

Fill Level	Pendulum Length (cm)	Fill Level	Pendulum Length (cm)
1%	10.09	60%	5.67
10%	9.47	70%	4.74
20%	8.77	80%	3.67
30%	8.05	90%	2.35
40%	7.30	99%	0.47
50%	6.51	——	——
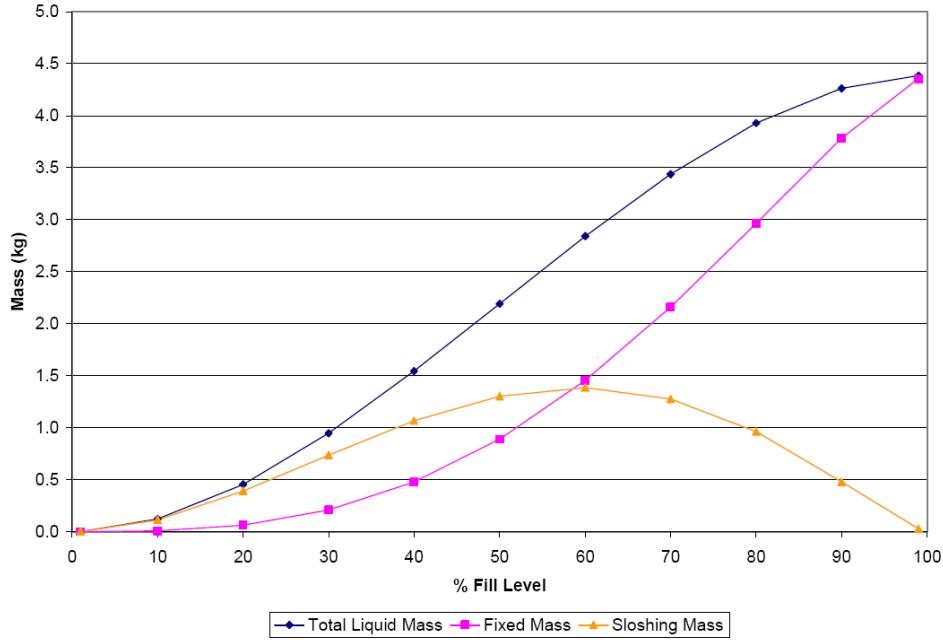
Figure 2.1   Liquid Mass Distribution for 8" Sphere [20]

Using this data, modelling fuel slosh in an eight inch sphere with a mechanical pendulum and fixed mass was validated against actual fuel slosh in a spherical glass tank. The constraint forces experienced by the tank involved in this study are much larger than those involved with SimSAT, so the fuel slosh in SimSAT should be smaller than what was seen in this article. However, the pendulum model used to describe fuel slosh is still applicable for use in the model of SimSAT.

Correctly modelling the behavior of liquids in space is important in relation to an accurate attitude dynamic model. Kezirian's article uses a different approach for fuel slosh modelling than the previous article [16]. Using fluid dynamics, he creates a computational fluid model that is then compared to the Liquid Motion Experiment performed on the Space Shuttle Atlantis on May 21-22, 1997 of NASA mission STS-84 [16]. This model is much too complex for use within this thesis, and it is modelling fluid movement in an environment independent of acceleration from Earth's gravity. The environment assumed by Kezirian's model is very different from the environment experienced by SimSAT, which would make the application of the model difficult even if the model was simple enough to implement in the Matlab® code of this thesis.

As the models for fluid dynamics in $0g$ environments become more accurate, the ability to accurately model a satellite's attitude dynamics increases drastically. Fuel slosh carries such a significant importance for the attitude dynamics because fuel usually accounts for a large portion of a satellite's weight prior to launch, and it remains influential through the majority of the satellite's lifespan. Without knowledge of the dynamics related to such a large percentage of a spacecraft's mass, accurate attitude dynamic models are difficult to create.

2.2 SimSAT Dynamic Model

In this research, the attitude control and the dynamics simulation of SimSAT have four parts: SimSAT dynamics, rotational dynamics, fuel slosh dynamics, and SimSAT's control law. First, the dynamics of the satellite must be determined using Euler's equations. Second, the dynamic relation between the satellite's motion and the Euler angles used to describe its position with respect to an inertial frame are developed. Then, the model used in this thesis for estimating fuel slosh is examined. Finally, the development of the control law for SimSAT is explained. The body axes used to describe SimSAT's dynamics are in Figure 2.2.
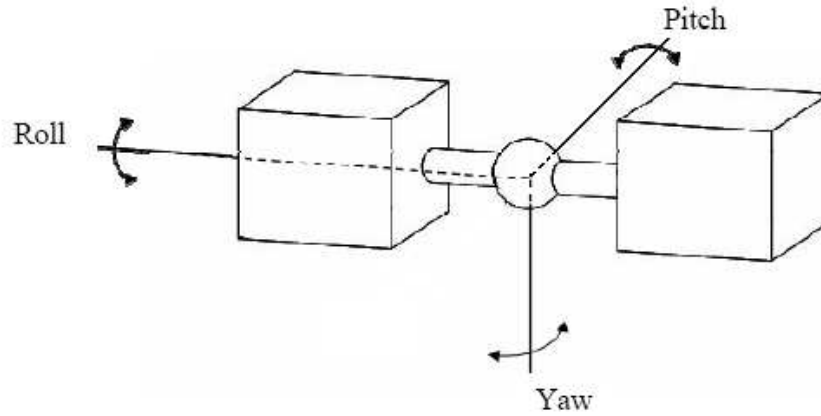
Figure 2.2   SimSAT Body Axes

2.2.1 Euler Equations. The majority of SimSAT's dynamics are determined from the basic Euler equations of motion:

$$M_{ext} = J_{SS} \cdot \dot{\omega} + \omega \times J_{SS} \cdot \omega \qquad (2.1)$$

where M_{ext} is a vector of the external moments about each body axis, ω is a vector comprised of SimSAT's angular rates about each body axis, and J_{SS} is the inertia matrix for SimSAT about the body axes. Equation 2.1 is mostly used as a simple model to describe how a free floating body in space will move in inertia space with respect to the defined body axes.

In order to use Equation 2.1 in describing SimSAT's general motion, the reaction wheels must be included in the system dynamics. In order to make this process both accurate and as simple as possible, a few assumptions are made:

1. The body axes are also the principal axes,

2. J_{SS} includes the inertia of everything on SimSAT including the reaction wheels when their relative angular velocities are zero, and

3. All reaction wheel rotation axes are aligned with a corresponding body axis.

Assumption 1 makes J_{SS} a three by three diagonal matrix. This assumption was made by Captain Dabrowski and Captain Geitgey. In addition, the coupling of the body rates was negligible during initial testing supporting the validity of assumption 1. Assumption 2 allows the inertia of the reaction wheels when they are at rest to be included with the rest of SimSAT, so now only the relative velocity of the reaction wheels (ζ_i) will have an affect on the dynamics of SimSAT beyond what is already modelled in Equation 2.1. Assumption 3 allows the torque from the reaction wheels to be included in the dynamics without rotation matrices.

To include the reaction wheels on SimSAT, the Euler equations must be derived to include the ability of the reaction wheels to change their angular momentum about their respective axes of rotation. SimSAT's angular momentum (H) with reaction wheels can

be described as:

$$H = J_{SS} \cdot \omega + J_{RW} \cdot \zeta_1 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + J_{RW} \cdot \zeta_2 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + J_{RW} \cdot \zeta_3 \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad (2.2)$$

where J_{RW} is the inertia of the reaction wheels about their respective rotation axes and ζ_i is the relative angular velocity of the reaction wheel about the ith axis. To simplify Equation 2.2, J_{RW} can be made into a three by three diagonal matrix where each diagonal entry represents the inertia about the axis of rotation for a single reaction wheel. Likewise, ζ can be expressed as a three by one vector where each individual entry signifies the relative angular velocity of a single reaction wheel with respect to SimSAT's angular velocity about its relative body axis. With these simplifications, the derivative of Equation 2.2 with respect to the inertial frame is then equal to the external moments experienced by the satellite as seen in Equation 2.3. Equation 2.3 is the governing equation used to model SimSAT's dynamics in this thesis.

$$\dot{H} = M_{ext} = J_{SS} \cdot \dot{\omega} + J_{RW} \cdot \dot{\zeta} + \omega \times (J_{SS} \cdot \omega + J_{RW} \cdot \zeta) \qquad (2.3)$$

2.2.2 Rotational Kinematics. Equation 2.3 describes how SimSAT's angular rates will vary over time, but the Euler equations do not describe SimSAT's attitude over time. Euler angles are used to describe the position of SimSAT's body frame with respect to an inertial frame. There is a direct relation between ω and how the Euler angles change over time.

Euler angles define the magnitude of a rotation about a specific axis required to align two separate coordinate frames. The three elementary rotations are described below as:

$$C_1(x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(x) & \sin(x) \\ 0 & -\sin(x) & \cos(x) \end{bmatrix} \tag{2.4}$$

$$C_2(x) = \begin{bmatrix} \cos(x) & 0 & -\sin(x) \\ 0 & 1 & 1 \\ \sin(x) & 0 & \cos(x) \end{bmatrix} \tag{2.5}$$

$$C_3(x) = \begin{bmatrix} \cos(x) & \sin(x) & 0 \\ -\sin(x) & \cos(x) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

where C_1 is the rotation matrix about axis one, C_2 is the rotation matrix about axis two, C_3 is the rotation matrix about axis three, and x represents an arbitrary angle to rotate about the designated axis.

The Roll-Pitch-Yaw rotation scheme is comprised of the three elementary rotations, which can be combined into a single rotation matrix capable of transforming the inertial frame to the body frame. Equation 2.7 shows how this rotation matrix is formed from Equations 2.4, 2.5, and 2.6.

$$C^{B/I} = C_3(\psi) \cdot C_2(\theta) \cdot C_1(\phi) \tag{2.7}$$

In this configuration ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. These equations now describe the transformation from the inertial frame to the body frame, but Equation 2.7 does not account for the movement of the body frame with respect to the inertial frame.

In order to determine the relationship between the angular rates in the body frame and the rate of change of the Euler angles, the derivatives of the Euler angles must be related to the body frame angular rates using rotation matrices. Using the Roll-Pitch-Yaw rotation scheme the relationship between the body angular rates and the Euler angles are

in Equation 2.8. Equation 2.8 can be solved for the derivatives of the three Euler angles seen in Equation 2.9.

$$
\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + C_3\left(\psi\right) \cdot \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + C_3\left(\psi\right) \cdot C_2\left(\theta\right) \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} \tag{2.8}
$$

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\left(\theta\right)} \begin{bmatrix} \cos\left(\psi\right) & -\sin\left(\psi\right) & 0 \\ \sin\left(\theta\right) \cdot \cos\left(\theta\right) & \cos\left(\theta\right) \cdot \cos\left(\psi\right) & 0 \\ -\sin\left(\theta\right) \cdot \cos\left(\psi\right) & \sin\left(\theta\right) \cdot \sin\left(\psi\right) & \cos\left(\theta\right) \end{bmatrix} \cdot \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{2.9}
$$

We can determine SimSAT's attitude at any time by integrating the derivative of the Euler angles over time along with an initial condition. From Equation 2.9, we can determine the current roll, pitch, and yaw angles and their rates describing SimSAT's attitude which can be used to orient SimSAT using feedback control [25]. The only remaining dynamics that need to be incorporated are the dynamics related to fuel slosh.

2.2.3 Fuel Slosh.   Fuel slosh is an important dynamic quantity to include in any analysis of satellite motion. Sloshing fuel is a source of energy loss and disturbance moments. The disturbance moment created by fuel slosh is an internal moment, so its net affect is zero once the fuel has reached equilibrium. The fuel's ability to decrease the total amount of energy in the satellite and to push the satellite around while moving to the point of least energy makes the fuel's movement a very important aspect that needs to be understood and modelled.

There are several methods to modelling fuel slosh using masses, springs, dampers, and pendulums. For the purpose of this thesis fuel slosh is modelled as a three dimensional pendulum with a moving support adapted from the article by Keith Schlee [20]. This system has five degrees of freedom: the three translational degrees of freedom for the moving support and two degrees of freedom for the angular motion of the pendulum bob. The differential equations used in this thesis for modelling fuel slosh with a 3D pendulum

came from the Lagrangian and its differential equations seen in Equation 2.10 and 2.11.

$$\mathcal{L} = T - V \tag{2.10}$$

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_k}\right) - \frac{\partial \mathcal{L}}{\partial q_k} = Q_k \tag{2.11}$$

Where Ł is the Lagrangian, T is the kinetic energy, V is the potential energy, q_k is the kth generalized coordinate used to describe the pendulum's motion and Q_k is the kth generalized force that provides the virtual work along the associated generalized coordinate [18] in Equations 2.10 and 2.11.

Figure 2.3 shows two of the five generalized coordinates used to describe the pendulum's position and motion. The translation of the support in not shown, but is along the IJK axes.
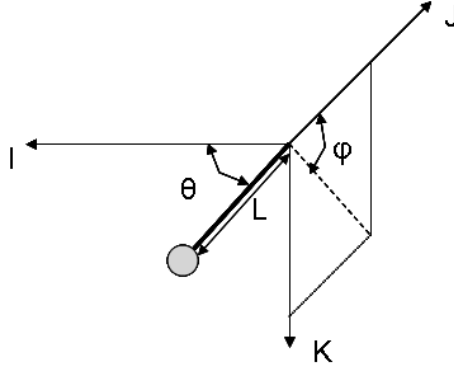
Figure 2.3   3D Pendulum Generalized Coordinates

The differential equations for the two generalized coordinates from Figure 2.3 are in Equations 2.12 and 2.13.

$$\ddot{\theta} = \frac{L \cdot \cos\left(\theta\right) \cdot \sin\left(\theta\right) \cdot \dot{\phi}^2 + \sin\left(\theta\right) \cdot \ddot{x} + \cos\left(\theta\right)\left(-\cos\left(\phi\right) \cdot \ddot{y} + \sin\left(\phi\right)\left(g - \ddot{z}\right)\right)}{L} \tag{2.12}$$

$$\ddot{\phi} = \frac{-2 \cdot L \cdot \cos\left(\theta\right) \cdot \dot{\theta} \cdot \dot{\phi} + \sin\left(\phi\right) \cdot \ddot{y} + \cos\left(\phi\right) \cdot \left(g - \ddot{z}\right)}{L \cdot \sin\left(\theta\right)} \tag{2.13}$$

While Equations 2.12 and 2.13 describe the two generalized coordinates a stationary pendulum requires, they do not include the motion of the pendulum's support. Equations 2.14, 2.15, and 2.16 account for the other three degrees of freedom describing the motion of the pendulum's support.

$$Q_x = -m \cdot \left(L \cdot \cos(\theta) \cdot \dot{\theta}^2 - \ddot{x} + L \cdot \sin(\theta) \cdot \ddot{\theta} \right) \tag{2.14}$$

$$
\begin{aligned}
Q_y = & -m \cdot (L \cdot \cos(\phi) \cdot \sin(\theta) \cdot \dot{\theta}^2 + 2 \cdot L \cdot \cos(\theta) \cdot \sin(\phi) \cdot \dot{\theta} \cdot \dot{\phi} \\
& + \quad L \cdot \cos(\phi) \cdot \sin(\theta) \cdot \dot{\phi}^2 - \ddot{y} - L \cdot \cos(\theta) \cdot \cos(\phi) \cdot \ddot{\theta} \\
& + \quad L \cdot \sin(\theta) \cdot \sin(\phi) \cdot \ddot{\phi})
\end{aligned} \tag{2.15}
$$

$$
\begin{aligned}
Q_z = & \; m \cdot (L \cdot \sin(\theta) \cdot \sin(\phi) \cdot \dot{\theta}^2 + 2 \cdot L \cdot \cos(\theta) \cdot \cos(\phi) \cdot \dot{\theta} \cdot \dot{\phi} \\
& - \quad L \cdot \sin(\theta) \cdot \sin(\phi) \cdot \dot{\phi}^2 + \ddot{z} + L \cdot \cos(\theta) \cdot \sin(\phi) \cdot \ddot{\theta} \\
& + \quad L \cdot \cos(\phi) \cdot \sin(\theta) \cdot \ddot{\phi})
\end{aligned} \tag{2.16}
$$

$\ddot{x}$, $\ddot{y}$, and $\ddot{z}$ are the translational accelerations along the inertial axes I, J, K respectively. Equations 2.14, 2.15, and 2.16 were adapted from Meirovitch, which provides examples for a stationary 3D pendulum and a planar pendulum with a moving base [18].

For the experimental setup of SimSAT, Equation 2.3 are combined with the equations of motion of the 3D pendulum by calculating the moments caused by the constraint forces at the pendulum's support. In order to do this, two assumptions have to be made. First, SimSAT's center of mass is not moving. Second, the vector from the center of mass to the support of the pendulum is known. Without these assumptions another dynamic model would have to be created in order to describe how the center of mass and the support of the pendulum move. Since SimSAT can be assumed to be a semi-rigid structure the dynamics of the two points will be negligible, so they are considered to be stationary with respect to SimSAT's body axes. With the two assumptions stated above the moment caused by fuel slosh is found with Equation 2.17:

$$M_{PEN} = \vec{r}_{PEN} \times \begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} \tag{2.17}$$

where M_{PEN} is a vector of the moments about a body axis and $\vec{r}_{PEN}$ is the displacement vector between SimSAT's center of gravity and the support of the 3D pendulum in the body frame. The moments from the pendulum are then added to Equation 2.3 as external moments, and it is now possible to model SimSAT with the fuel slosh dynamics included.

2.3   SimSAT Controller

Feedback control is commonly used in order to maintain a specific orientation. In this thesis, dynamic inversion was used in order to remove the majority of the nonlinearities inherent in a satellite's attitude dynamics and replace them with desired dynamics [21]. This technique is also referred to as feedback linearization. When designing the controller for SimSAT, we assumed the external moments were zero and the angular momentum of the reaction wheels are much smaller than the angular momentum of SimSAT in order to simplify the control law. These assumptions should not be used when attempting to model the dynamics of SimSAT, but they are good enough to calculate a control law capable of basic control of SimSAT. With these assumptions, Equation 2.3 can be made into the form:

$$J_{SS} \cdot \dot{\omega} + \omega \times J_{SS} \cdot \omega = u \tag{2.18}$$

where u is a vector comprised by the input torques from the three reaction wheels. In order to remove the nonlinear component of this equation, u must take the form:

$$u = \omega \times J_{SS} \cdot \omega + J_{SS} \cdot \nu \tag{2.19}$$

where ν is the desired dynamics. For this thesis, the desired dynamics are:

$$\nu = k_p \cdot \theta_{error} + k_d \cdot \omega \tag{2.20}$$

where θ_{error} is the error in the Euler angles, k_p is the proportional gain, and k_d is the derivative gain. The proportional gain (k_p) weights the affect of the error between the desired and measured Euler angles on the control input. Additionally, the derivative gain (k_d) weights the affect of the angular velocity of SimSAT on the control input. Finally,

the control law for SimSAT becomes:

$$u = \omega \times J_{SS} \cdot \omega + J_{SS} \cdot (k_p \cdot \theta_{error} + k_d \cdot \omega) \tag{2.21}$$

J_{SS} can be incorporated into the controller gains (k_p and k_d) for the second part of the equation dealing with the Euler angle error and the body rates. For the first part of the equation used to remove the nonlinearities associated with the coupling of the body rates in the dynamics, J_{SS} can be approximated as 5 $kg \cdot m^2$ for the roll axis and 60 $kg \cdot m^2$ for the pitch and yaw axes. These values are approximations found from initial experimentation attempting to estimated SimSAT's MOI. The resulting control law's Simulink$^{®}$ diagram is seen in Figure 2.4.
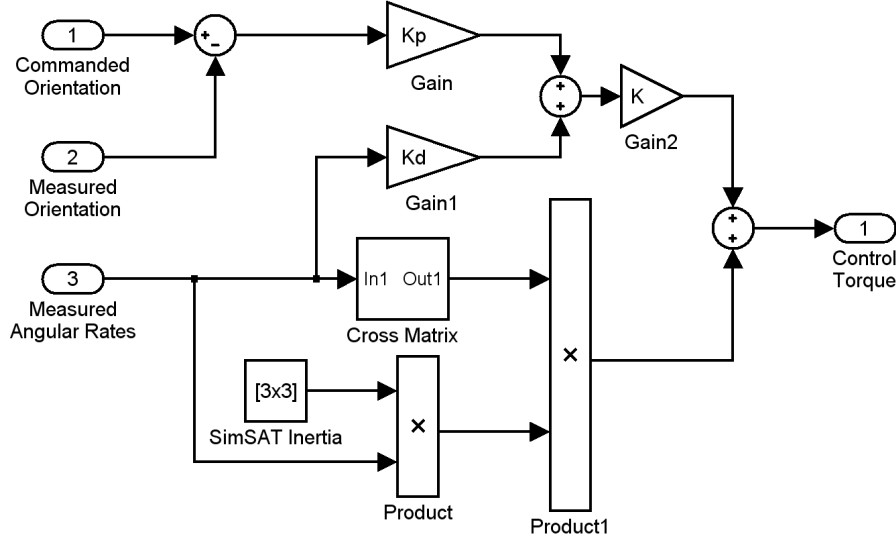
Figure 2.4   SimSAT's Control Law Simulink$^{®}$ Diagram

where K is an overall gain. The values for SimSAT's controller used in this thesis can be found in Table 2.2. The controller values were found through experimental runs with real time tuning, and were chosen in order to create a quick response with little overshoot.

The controller diagramed in Figure 2.4 with the controller gains described in Table 2.2 is designed to keep the angular velocities about the roll and pitch axes at zero while the Euler angle describing SimSAT's position in the yaw axis is used for closed-loop pointing

Table 2.2   SimSat Controller Parameters

Parameter	Value
Kp yaw	30
Kp pitch	0
Kp roll	0
Kd yaw	-110
Kd pitch	25
Kd roll	15
K	-1

control. The negatives for the specific axes are based on the relative pointing direction of the reaction wheels rotation axes compared to the body axes. In addition, the torque experienced by the flywheel from the motor will be the negative of that experienced by SimSAT from the motor. The roll and pitch reaction wheels are used mainly for disturbance torque cancellation, which is used later to develop a working model of SimSAT's dynamics.

2.4   Estimation Process Overview

There are two estimation processes used in this research. One processes provides an estimate for SimSAT's yaw MOI. The other processes uses two estimates of MOI in order to estimate the change in fuel mass. The components of the MOI estimation are outlined in Section 2.4.1, while the components for the change in fuel mass estimation are outlined in Section 2.4.2. The statistical methods used to remove outlier data from data sets for the MOI estimation process is in Section 4.4.

2.4.1   MOI Estimation.   The core of the fuel estimation process is the Matlab® script fmincon. The fmincon script will find the values of specified variables that minimize a given cost function that is dependent on those variables. According to the Matlab® help file for the fmincon script, "fmincon attempts to find a constrained minimum of a scalar function of several variables starting at an initial estimate. This is generally referred to as constrained nonlinear optimization or nonlinear programming." The fmincon script solves the optimization problem described below [1]:

$$min_x \ f(x) \text{ subject to}$$
$$c(x) \leq 0$$
$$ceq(x) = 0$$
$$A \cdot x \leq b$$
$$Aeq \cdot x = beq$$
$$lb \leq x \leq ub$$

where x is a vector comprised of the dependent variables, $f(x)$ represents the scalar value of a function evaluated at the dependent variables, $c(x)$ are the nonlinear inequality constraints, $ceq(x)$ are the nonlinear equality constraints, $A \cdot x \leq b$ describe the linear inequality constraints in matrix from, $Aeq \cdot x \leq beq$ describe the linear equality constraints in matrix form, lb are the lower bounds of the dependent variables, and ub are the upper bounds of the dependent variables. Using an initial guess of the dependent variables as a starting point, the fmincon script uses a search algorithm to find the values of the dependent variables that decrease the scalar value of the cost function. The fmincon script is an iterative process that continually searches for better values for dependent variables until designated requirements are met, and the resulting values can then be considered at least as local minimum for the cost function.

This thesis uses fmincon to find the MOI of SimSAT about the yaw axis. The yaw MOI is estimated by using fmincon to minimize the error between a model of SimSAT's dynamics and the measured angular body rates. The only constraints used are upper and lower bounds on the dependent variables of the model. The MOI estimation processes is outlined in the flow chart in Figure 2.5.

2.4.1.1 SimSAT Model. Fmincon requires a working model of SimSAT in order to estimate the yaw MOI. Dabrowski and Geitgey, the previous two students, created models that used the yaw angle commanded for an input and provided simulated yaw angles over time to be compared to experimental angles from SimSAT [11, 14]. For reasons discussed in Section 4.3.2, the model for this research uses the reaction wheels' measured angular velocities as inputs and provides body angular rates to be compared with the measured angular rates from SimSAT.
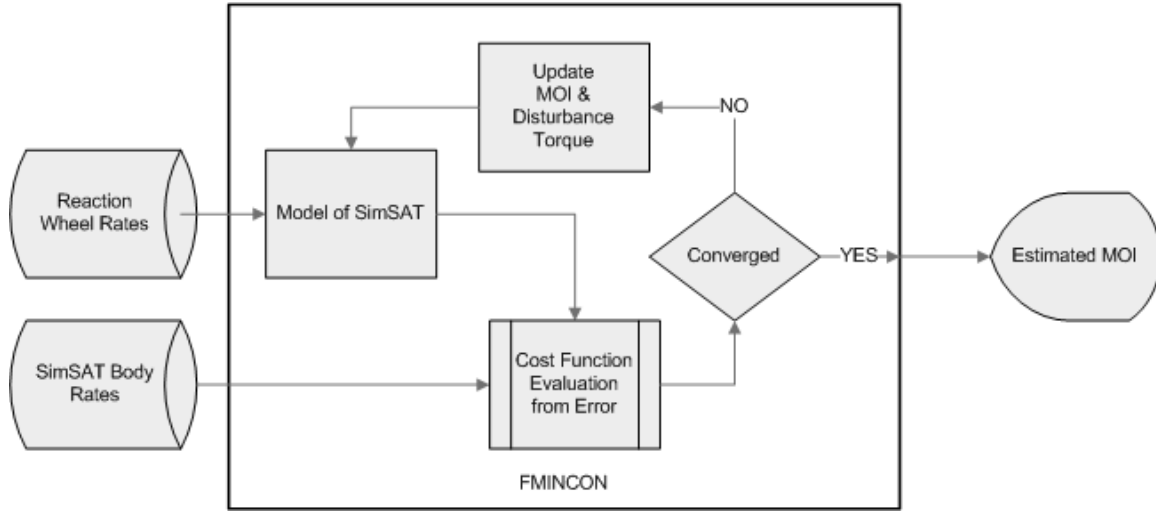
Figure 2.5   MOI Estimation Flow Chart

The model of SimSAT allows fmincon to search for the MOI and disturbance torque about the yaw axis that minimizes the specified cost function. The Matlab® and Simulink® code used to find the estimates of MOI and disturbance torque are in Appendix C. The fmincon script finds the MOI and disturbance torque that minimizes the error between the modelled SimSAT body rates and the measured rates. The resulting MOI estimations are dependent on the form of the cost function used within the fmincon script. In order to test the sensitivity of the MOI estimation process to the cost function, three different cost functions were examined.

2.4.1.2 Cost Functions.   There were three different cost functions that fmincon minimizes in this thesis in order to ascertain the general effect the cost function would have on the process. The first cost function equally weights the components of the error vector from the difference between the actual SimSAT yaw rate data and the simulated yaw rate. The second and third cost functions weight the components of the error vector based on SimSAT's measured yaw rate at the time corresponding to the error entry. If the absolute value of the measured yaw rate of SimSAT was greater than 0.005 rad/sec the corresponding error at that time was weighted heavier than when it was less than 0.005 rad/sec. A mathematical representation of the cost function is in Equation 2.22. The form of the cost function was chose for simplicity in the implementation within

Matlab®. Matlab® has a norm function that performs the two norm on a vector in order to get a representative scalar value of the vector.

$$Cost = \sqrt{\sum_{i=0}^{k} \left[w(t_i) \cdot \left(\dot{\psi}(t_i)_{measured} - \dot{\psi}(t_i)_{simulated} \right) \right]^2} \qquad (2.22)$$

In Equation 2.22 $w(t_i)$ is the weight assigned at that specific time, $\dot{\psi}(t_i)_{measured}$ is the measured angular yaw rate, $\dot{\psi}(t_i)_{simulated}$ is the simulated angular yaw rate, and k is the number of measurements taken in one test found by dividing the final time by the internal time step (0.05 seconds) of the Simulink® model used to control SimSAT.

The weighting for Cost Function 1 is described mathematically by Equation 2.23.

$$w(t_i) = 1 \ \forall \ t \qquad (2.23)$$

Cost Function 1 suggests that the points where SimSAT is not moving has as much importance to the MOI estimation process as the time when SimSAT is changing orientation. MOI estimation is done by studying the relationship between a known torque and the resulting change in the angular rate of the body. When SimSAT is not moving the control torque is of the same magnitude as the disturbance torques making the MOI estimation process more sensitive to variations in the disturbance torque that are not modelled. The second cost function weights all of the data points that correspond to SimSAT moving faster than 0.005 rad/sec in either direction about the yaw axis as twice as important as the points where SimSAT is not. A mathematical representation of the weighting convention for Cost Function 2 is described by Equations 2.24a–b.

$$w(t_i) = 1 \ \forall \ t_i \ \text{when} \ \left| \dot{\psi}(t_i)_{measured} \right| < 0.005 rad/sec \qquad (2.24a)$$

$$w(t_i) = 2 \ \forall \ t_i \ \text{when} \ \left| \dot{\psi}(t_i)_{measured} \right| \geq 0.005 rad/sec \qquad (2.24b)$$

The value of 0.005 rad/sec was chosen because that is the smallest speed that avoids the vast majority of the measurement noise when SimSAT is not moving. Likewise, the third cost function weights the same points as five times as important. The weighting convention

2-16

for Cost Function 3 is described by Equations 2.25a–b.

$$w(t_i) = 1 \ \forall \ t_i \text{ when } \left| \dot{\psi}(t_i)_{measured} \right| \ < \ 0.005 rad/sec \tag{2.25a}$$

$$w(t_i) = 5 \ \forall \ t_i \text{ when } \left| \dot{\psi}(t_i)_{measured} \right| \ \geq \ 0.005 rad/sec \tag{2.25b}$$

The weighted error vector is composed of the difference between the measured and the modelled angular rates of SimSAT at a specific time multiplied by its corresponding weighting of 1, 2, or 5. The cost that is passed to the fmincon script is then the two norm of the weighted error vector as seen in Equation 2.22. Once `Matlab`® has minimized this cost, the estimated yaw MOI for the measured data from the test run has been found.

2.4.2 Fuel Estimation. From two estimated MOI values the change in fuel mass can be estimated. Several assumptions were made in order to make this fuel estimation technique work. First, the water mass is not moving in the tanks. The reason for this assumption will be discussed in Section 4.3.2. Since the water is not moving, the MOI of the water about the yaw axis can be approximated using the parallel axis theorem as seen in Equation 2.26

$$I_b = I_a + m \cdot d^2 \tag{2.26}$$

where I_a is the MOI of the water about the original axis, I_b is the MOI about the axis that is parallel from the original, m is the total mass of the water, and d is the perpendicular distance between the two parallel axes. In addition, we assume that for a specific configuration SimSAT's MOI will not change unless the mass properties of SimSAT are changed. For the purpose of this thesis, we assume that the only change in the mass properties of SimSAT come from the change in the amount of water in SimSAT's tanks. In other words, SimSAT is a rigid body, meaning it has no flexing or moving parts, and no extra equipment was added to SimSAT between tests.

In this thesis, a baseline MOI of SimSAT will be calculated where there is no water in the tanks. Then, test MOIs will be estimated with different amounts of water in the tanks. The mass of water is then estimated using Equation 2.27 which is derived from

Equation 2.26.

$$m_{water} = (I_{estimated} - I_{baseline} - I_{water})/d^2 \qquad (2.27)$$

In Equation 2.27 $I_{estimated}$ is the estimated MOI of SimSAT with a known amount of water in the tanks, $I_{baseline}$ is the estimated MOI of SimSAT without water in the tanks, and I_{water} is the MOI of the water about an axis parallel to the yaw axis that passes through the center of gravity of the water in the tanks. The calculation of I_{water} is dependent on the mass of the water in the tanks, so this dependence requires an iterative process to be used to solve Equation 2.27 for the estimated water mass.

The inertia's dependence on the mass of water in the tanks is complex. The axes and dimensions used to describe this relationship are in Figure 2.6. With the mass of water in one tank (half of the estimated water mass) and water's density the volume of water in a tank is found. The fill height of water in the tank is then related to the volume of water by Equation 2.28 where h is the z value corresponding to the fill height of the tank. The inertia of the water in one tank is then described by Equation 2.29 [3].

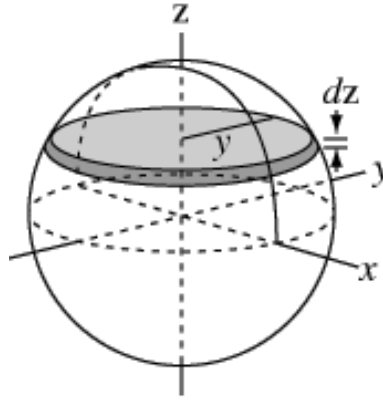
Figure 2.6   Spherical Dimensions [3]

$$V = \pi \int_{-R}^{h} y^2 dz \text{ where } y^2 = R^2 - z^2 \qquad (2.28)$$

$$I_{water} = \frac{1}{2} \cdot \rho \cdot \pi \int_{-R}^{h} \left(R^2 - z^2\right)^2 dz \qquad (2.29)$$

Initially, Equation 2.27 is used with I_{water} equal to zero. With half of the resulting estimated value for the water mass the inertia of the water in one tank about the z axis is calculated with Equations 2.28 and 2.29. The inertia of all the water is then twice that of the calculated inertia for the water in one tank. Upon witch Equation 2.27 is reevaluated with the calculated I_{water}. The iterations of this process continue until the changes in the inertia of the water about the z axis are reduced to the point of insignificance. Once convergence has been achieved the estimate for the water mass has been found. The code used for the iterative process is in Appendix C.7.

2.4.3 Outlier Data Analysis Techniques. When the model for SimSAT is unable to accurately portray reality the result of the MOI estimation process is not accurate. When these bad results are grouped with MOI estimates from good test runs, the bad results appear as outliers. The bad data points need to be identified and removed to increase the accuracy of the MOI estimation processes. For this research, the Grubbs Test will be used to identify the estimated MOI values that can be considered outliers. The Grubbs test was chosen because it can be made into an automated processes which increases the repeatability of results from the MOI estimation process.

The Grubbs Test was developed by F.E. Grubbs, and is a well used technique for identifying outliers. The first step is to rank order all of the data points from lowest to highest. Next, calculate T according to Equations 2.30 and 2.31, where $\overline{X}$ is the mean of the data set, X_1 refers to the smallest value in the data set, X_n refers to the largest value in the data set, and s is the standard deviation of the data set. Equation 2.30 is used to decide if the smallest value in the data set is an outlier. Equation 2.31 is used to decide if the largest value in the data set is an outlier.

$$T_{low} = (\overline{X} - X_1)/s \tag{2.30}$$

$$T_{high} = (X_n - \overline{X})/s \tag{2.31}$$

Once T_{low} and T_{high} have been calculated, they must be compared to the associated critical value for the desired level of confidence [23]. The critical value for the Grubbs test

2-19

is calculated using Equation 2.32, where N is the number of data points in the set and t is the critical value of the t distribution with a dimension of $N - 2$ and a confidence interval decided on by the user.

$$T_{crit} = \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2}{N - 2 + t^2}} \tag{2.32}$$

In this thesis a confidence interval of 95% was used, which means that the Grubbs test will only identify a data point as an outlier if their is at least a 95% chance that the data point is bad. There is a 5% or less chance that the data points that are removed using the Grubbs test were actually viable data. The calculated critical value then needs to be compared to the calculated value for the smallest and largest data points in data set. If the critical value is less than the calculated value that point can be removed. When both the smallest and largest data point qualify to be removed, remove the point that has the largest difference between the calculated and the critical T value. Once a single point is removed, the test begins again at the beginning with the data set minus the outlier. This process repeats until there are no remaining outliers [2].

III. Experimental Equipment

SimSAT was designed and built by Colebank, Jones, Nagy, Pollak, and Manebach in 1999. An in depth description of SimSAT's original setup can be found in their thesis [10]. Since then, SimSAT has had several upgrades as technology has improved and been made available. SimSAT's purpose has always been to simulate on-orbit characteristics of a satellite in order to allow hardware validation of theoretical principles. Before this thesis, SimSAT appeared as it does in Figure 3.1. SimSAT consists of four main parts: the ground station, the air pedestal, the basic onboard systems, and other experimental equipment.
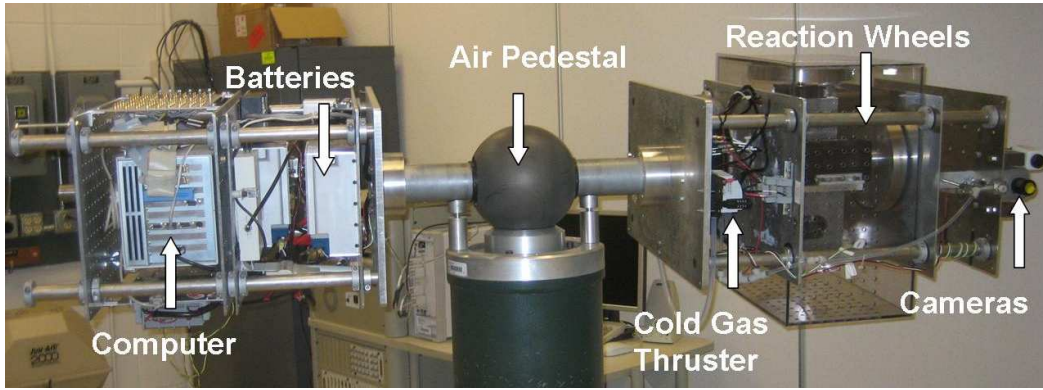
Figure 3.1   SimSAT Appearance Prior to this Thesis

3.1   Hardware Specifications

3.1.1   Ground Station.   The purpose of the ground station is to be the conduit of information between the user and SimSAT. For this purpose, the ground station is comprised of a desktop computer and a wireless router. Currently the ground station is run on a Dell Demension 4500 with Control Desk and Simulink®. The Simulink® models are created to process data from individual instruments and combine them into a working satellite hierarchy. Control Desk then converts these Simulink® files with a compiler into a format that the onboard computer understands. In addition Control Desk sets up the data links between the ground station and SimSAT's onboard computer. This allows the operator to interact with SimSAT on a real time basis through the wireless connection

provided by the RadioLan wireless DockLINK model 408. The RadioLan operates at a frequency of 5.8 GHz with data transfer speeds up to 10 Mbps.

3.1.2 Air Pedestal. In order to simulate the zero gravity environment of a satellite, the air pedestal creates a thin layer, less than $12.7\mu m$ thick, of air between the ball bearing at the center of the satellite and the top of the pedestal. This provides a force equal to the weight of the satellite in as near frictionless method as possible under the circumstances. Table 3.1 contains all of the air bearings pertinent specifications.

Table 3.1   Tri-Axis Spherical Air Bearing Specifications

Parameter	Value	Units
Outside Diameter of Rotor Sphere	8.66	in
Nominal Centerline Height	56	in
Rotor Weight	42	lb
Payload Weight Capacity	300	lb
Moment Capacity	6000	in-lb
Bending Stiffness	8.6×10^6	in-lb/rad
Access Hole Diameter	2.25	in
Yaw (Rotation about the Vertical Axis)	360	°
Roll (Rotation about the Horizontal Shaft)	360	°
Pitch (Tilt in Vertical Plan)	±30	°

3.1.3 Onboard Systems. The onboard computer that runs SimSAT and the user supplied programs is a dSpace AutoBox DS400. It coordinates and processes all of the data throughout SimSAT. All onboard electronics are powered by three 12 volt batteries that are combined in such a way that power can be supplied at 12, 24, and 36 volts.

One of the many systems that the computer controls is the attitude control system. The orientation of a satellite relevant to a reference frame, usually an inertial frame, is considered a satellite's attitude. In many satellites the mission relies on the satellite having a specific attitude. For example, communication devices must be pointed at Earth based ground stations in order to transmit data back to Earth. If the satellite had the wrong attitude the communication devices might be aimed at the moon causing the satellite to

be inoperable. SimSAT's attitude control system is comprised of a gyro and three reaction wheels.

       3.1.3.1 Gyro.   A gyro is an instrument capable of measuring the body angular rates and/or accelerations of a satellite relevant to an inertial frame. SimSAT has a Northrop Gurmman LN-200 Fiber Optic Gyroscope seen in Figure 3.2. Table 3.2 displays all of the Gyro's pertinent information.

Figure 3.2   LN-200 Fiber Optic Gyroscope

Table 3.2   Northrop Gurmman LN-200 Gyro Specifications

Parameter	Value	Units
Weight	700	g
Diameter	8.9	cm
Height	8.5	cm
Power Consumption	10	W
Bias Repeatability	$1-10$	/hr
Random Walk	$0.04-0.1$	$^\circ$ hr$^{1/2}$ power spectral density
Data Latency	<1	msec
Data Protocol	RS-485	
Data Structure	Synchronous Data Link Control (SDLC)	

With this gyro it is possible to keep track of SimSAT's attitude relative to an inertial frame comprised of its body frame's initial position. The gyro is connected with the

onboard computer by the LN-200 interface board. This board converts the constant data stream from the gyro into 21 byte data packets that are then sent to SimSAT's onboard computer for processing.

3.1.3.2 Reaction Wheels.   The reaction wheels are comprised of two parts: the flywheel and the motor. The flywheel is an aluminum plate with a steel rim around its edge. The wheel weighs 2.040 kg and has an inertia about the axis of rotation of $1.955E-2$ $kg \cdot m^2$. The motor is an Animatic Smart Motor model SM3450, which is normally used for conveyer belt assemblies. The motor's pertinent information can be seen in Table 3.3.

Table 3.3   *SM*3450 Motor Specifications

Parameter	Value	Units
Continuous Torque	250	oz-in
Peak Torque	750	oz-in
Top Speed	3398	rpm
Rotor Inertia	$2.12e-4$	kg-m^2

3.1.4 Specialty Items.   Previous students have added a cold gas thruster with a paintball compressed air tank along with cameras and other specialty items needed for their thesis work. For this thesis, two spherical tanks were created using AFIT's three dimensional printer and attached to each end of SimSAT. Each sphere has an inner diameter of eight inches in order to match the tank specifications in the article Parameter Estimation of Spacecraft Fuel Slosh [20]. There are two holes in the sphere. The top hole can be plugged and unplugged for filling the tank, and the bottom hole has a valve for easy and measurable draining. This allows the tanks to remain on SimSAT instead of removing them each time the fill level needs to be decreased. For this research, SimSAT appears as it does in Figure 3.3.

Figure 3.3   SimSAT in Final Configuration

3.2 SimSAT Preparatory Work

This section outlines the work done on SimSAT in order to get everything working correctly and accurately. The more recent upgrades to the gyro and the reaction wheels caused many problems when they were not properly integrated with SimSAT.

3.2.1 Gyro.

The previous SimSAT user showed that the fiber-optic gyro on SimSAT was malfunctioning on a regular basis [14]. In order to troubleshoot this problem, a systematic approach was taken to rule out possible causes. The data became corrupted after 500 seconds as shown in Figure 3.4. The data corruption problem would occur at random time intervals anywhere between 0 and 120 seconds. There were very few occasions where the data stream remained uncorrupted beyond 2 minutes. Figure 3.4 is an example of one of those times. The figure was used because it is the only saved data set available that shows the gyro worked properly for more than a second prior to the data becoming corrupted. Figure 3.4 also shows momentary points of data corruption.

The first step taken to find the source of this problem was to thoroughly examine the Simulink® code used to convert the raw data from the gyro into useful rate and acceleration data. There is not a way to request a data packet from the LN200 interface board, so they must be recorded as they are received. It is then imperative to identify the newest complete data set when the SimSAT software requires it. After studying the Simulink® code, we discovered that the correct twenty-one bytes were being pulled from the data, but the program continued to store and process an extra 20 bytes that were
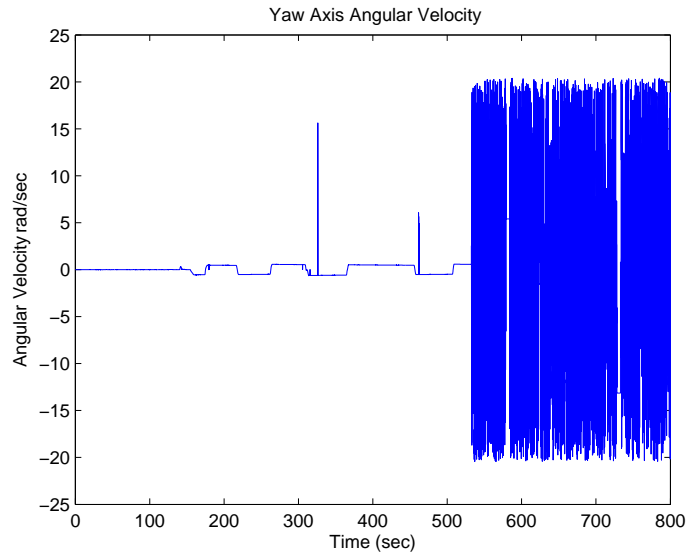
Figure 3.4   Yaw Angular Rate from Gyro with Corrupted Data

no longer useful. Removing the extra data bytes did not noticeably increase SimSAT's performance, but fewer calculations and stored data is always desired when the system has a limited amount of computing power and a lot of computing to do.

After confirming that the Simulink® code was correctly processing the raw data, the raw data itself was examined for a pattern related to the data corruption. We found that the third byte of the twenty-one bytes used is always 254 when the data was uncorrupted and something different once the data was corrupted. This showed that the data corruption was occurring before it ever made it to the Simulink® code.

This led to the examination of the LN-200 interface board. It is the board that is directly plugged into the gyro, and it is responsible for creating the twenty-one byte data packages sent to the onboard computer. The board converts the $12V$ power into $\pm 5V$ and $\pm 10V$. It turns out that the -5 volt power converter on the board was malfunctioning. Every time the data became corrupted, the -5 volt power on the board would register as 0 volts on the voltmeter. Once this was discovered as a possible source of the problem, a replacement LN-200 interface board was installed.

Following the installation of the new interface board several rounds of tests were conducted to make sure that the gyro was working properly. The overall process was

3-6

improved drastically, but two more problems were discovered. While the output from the gyro data no longer became a constant stream of corrupted data, there were isolated data points that were corrupted. In addition, the gyro data began to remain constant for about two seconds every two to three minutes. SimSAT's angular velocity was still changing during the period where the gyro data froze, so the controller then begins to provide increasing control inputs until the gyro data unfreezes and proper control resumes.

A source for the isolated data corruptions was not found, but the majority of the corrupted data values can be filtered out by replacing the corrupt data with the last good data package. This holds the angular rates coming from the gyro constant over two time steps, but the error caused by replacing bad data is much smaller than what would be caused by using the corrupt data. In Figure 3.5, the left plot shows gyro data converted without a filter to take out corrupted data points, and the right plot shows the same gyro data converted with the new filter.
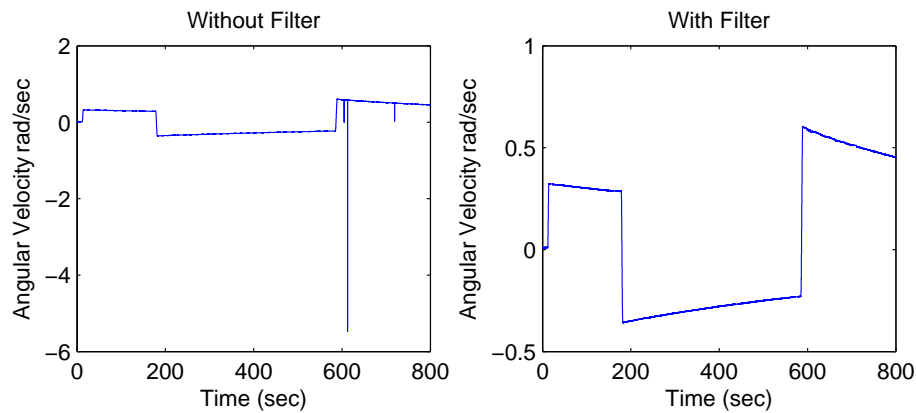

Figure 3.5   Yaw Angular Rate from Gyro

The freezing of the gyro data was never solved. When the gyro was attached directly to the computer this problem did not occur, so the problem source is assumed to be within the onboard computer. Since the data processing and memory storage techniques used by the onboard computer were not known, efforts were deemed not cost effective to further explore the problem since this problem could be filtered out of the final data. In addition, the SimSAT controller is able to bring SimSAT back under control shortly after the gyro freeze occurs.

3.2.2   Reaction Wheels.   Initially, SimSAT's reaction wheels had motors that were an analog version of the current models. The old motors were replaced with the digital model because an analog motor failed and the company no longer made them. It was believed that they were completely integrated with SimSAT. The initial work done with the reaction wheels was an attempt to characterize the reaction wheels since this was identified by previous students and professors as an important step to increasing the accuracy of the SimSAT model.

3.2.2.1   Characterization.   The goal of characterizing the reaction wheels on SimSAT was to create a lookup table that related the current velocity and the commanded change in velocity to the torque experienced. The characterization was done using a torque stand that was modified from the past design of a previous student and an extra reaction wheel. The torque stand was set up as seen in Figure 3.6.
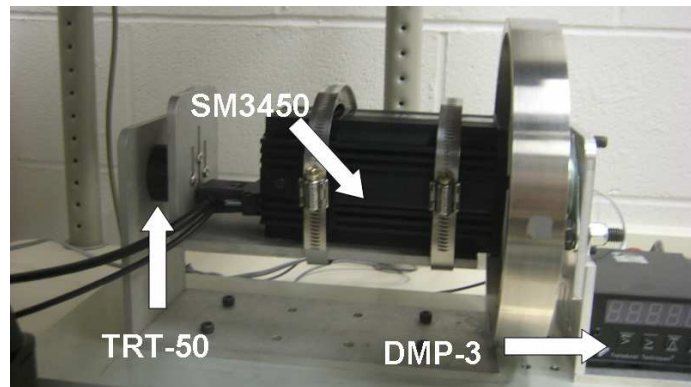

Figure 3.6   AFIT Torque Stand

The torque stand was made at AFIT, and used a Transducer Techniques TRT-50 torque sensor and a DPM-3 meter. The torque sensor's pertinent data can be seen in Table 3.4, and the sensor can be seen attached to the torque stand in Figure 3.6.

The DPM-3 meter's pertinent information is in Table 3.5, and the sensor can be seen next to the torque stand in Figure 3.6.

There are two modes for the reaction wheel motors that were studied: the velocity mode and the torque mode. The velocity mode has an internal controller that brings the angular velocity of the wheel to the commanded speed. The command for the reaction

3-8

Table 3.4   Transducer Techniques TRT-50 Sensor Specifications

Parameter	Value	
Capacity	50	in-lb
Max Shear	20	lb

Table 3.5   Transducer Techniques DPM-3 Meter Specifications

Parameter	Value
Accuracy - Load Cell Meter	0.01% FS $\pm$1 Ct.

wheel is then the desired angular velocity of the motor shaft. The torque mode deals directly with the amount of power provided to the motor with no controller. If the motor had a large resistive force to counter the commanded input would be the desired constant torque to be applied to the motor shaft. The setup for SimSAT does not have this resistive force, so the command is then proportional to the power given to the motor resulting in a steady state angular velocity achieved for the specific power setting.

Initially, the velocity mode was used. The measured torque from commands were jagged and noisy because of the internal controller. The internal controller was initially designed for assembly line applications, so it was not well suited for use as a reaction wheel. The torque mode bypasses this controller, which is why it was explored as an alternative option. At first glance the torque mode produced much cleaner torque curves from step commands. An example response can be seen in Figure 3.7.

Though the data initially looked promising, in the end, we determined that the torque stand was unable to produce results that were accurate or easily implementable in the SimSAT software. After running many tests and averaging the results together it was apparent that the data from the torque stand did not have the required accuracy to make it useful. Figure 3.8 shows data from three runs that started with an initial command of zero and tried to characterize the reaction wheel's response to changes in the command (Delta T).
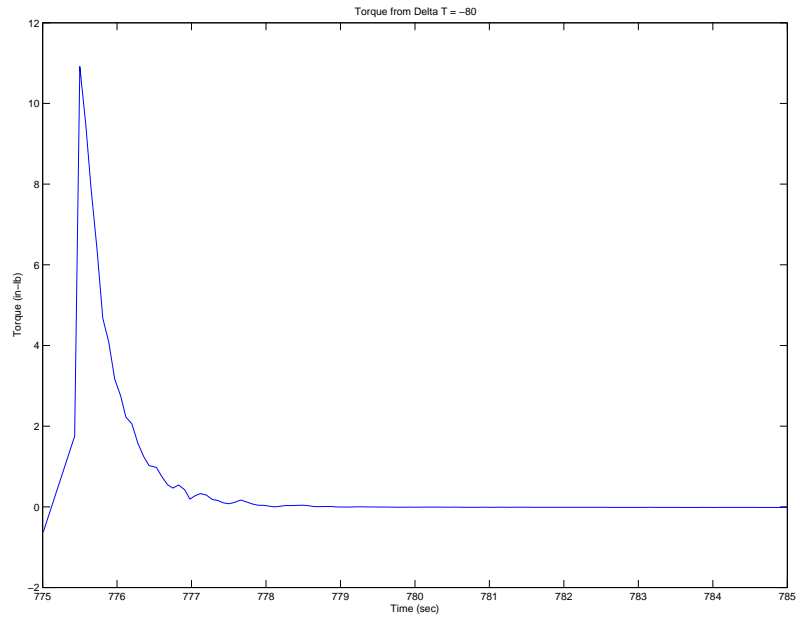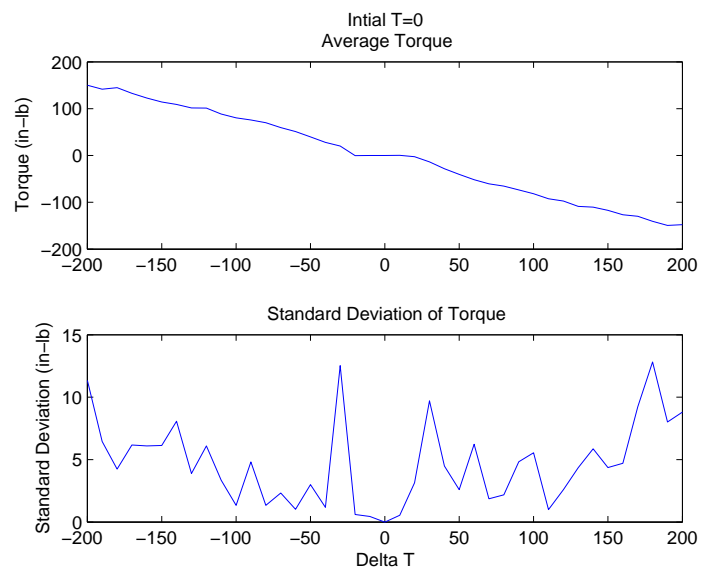
Figure 3.7   Torque Mode Example Response

Figure 3.8   Torque Mode Response Statistics

There is an obvious dead band around the zero point, which would need to be characterized
with more accuracy. This dead band has a nonlinear effect on the results when the initial
command is non-zero but the change in command (Delta T) brings it into the dead band

range. Most importantly the standard deviation seen in the bottom plot shows standard deviation values for each measured torque value. Standard deviations of 10 $in \cdot lb$ is large enough to cause serious errors once this data is integrated into the SimSAT architecture.

In the end, the preliminary characterization of a limited range of commands is seen in Figure 3.9. This surface shows the results of averaging 3 runs for each initial command
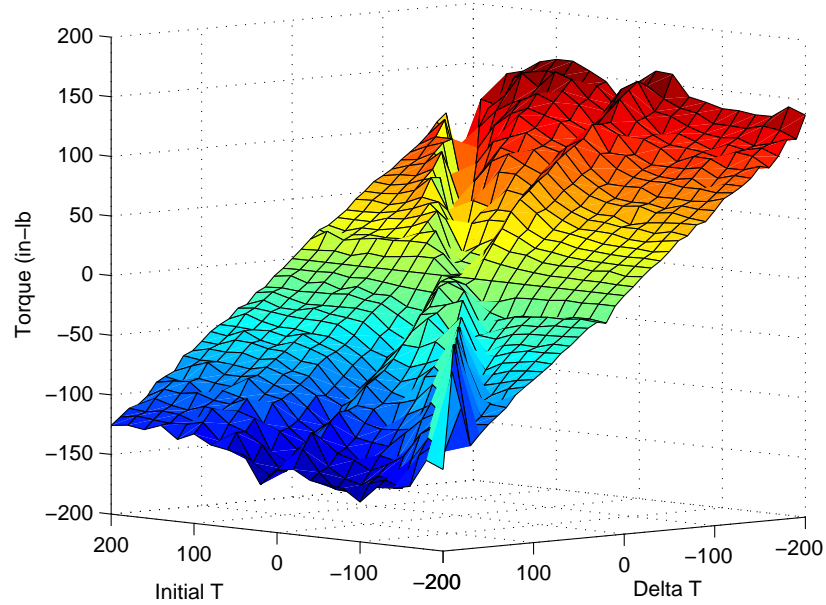

Figure 3.9   Torque Mode Lookup Table Depiction

with the delta commands. Even if the standard deviation of this method was much smaller, the SimSAT software only allows a lookup table to have 10000 total entries, which would not be enough entries to accurately characterize the reaction wheels over their operational range. In the end, the velocity mode was decided on as the best method to control the reaction wheels since it has the ability to directly control the reaction wheels' angular velocities even if the internal controller is ill-suited for the application.

3.2.2.2   Reaction Wheel Control.   It was desired to command the reaction wheel using a commanded torque rather than angular velocity in order to simplify SimSAT's controller. The lookup table from wheel characterization turned out to be impractical, so another method was used. A new Simulink® subfunction was created in order

3-11

to ideally calculate the desired reaction wheel angular velocity from the current angular velocity and the desired control torque. This is done using Equation 3.1.

$$\zeta_{command} = \frac{\Delta t \cdot Torque}{I_{RW}} + \zeta_{current} \qquad (3.1)$$

ζ is the reaction wheel's relative angular velocity, Δt is the time step of the Simulink$^{\circledR}$ program, and I_{RW} is the moment of inertia of the reaction wheels about their axes of rotation. For this to work accurately, the desired torque must be kept reasonably small in order to ensure the reaction wheels are able to accurately track the resulting angular velocities. In addition, Δt must be as small as possible, and is 0.05 for this thesis. It is a small enough time step that SimSAT's operation seems continuous, and the onboard computer has enough computational power to keep up with it.

 3.2.2.3 Motor Tuning.   The SM3450 motors use an internal controller when they are in velocity mode. Since this controller was originally designed for a conveyer belt application, its use for a reaction wheel motor on SimSAT is difficult at best. The exact diagram of the controller is unknown due to proprietary reasons. A tuning utility is provided by Animatics, but it was never designed to handle an application without a dominant resistive force. Therefore, tuning the controller for the reaction wheels must be done by hand. The motor parameters used in this thesis are in Table 3.6.

Table 3.6   Reaction Wheel Motor Tuning Values

Parameter	Value
KP	13
KI	0
KL	0
KD	3500
KV	10
KA	5
A	130
E	3200000

It is possible to use step inputs with a magnitude greater than 200 radians per second without encountering a limit cycle. The previous controller parameters entered limit cycles on a regular basis, so while not ideal, the current parameters are a large improvement. SimSAT now has full use of the reaction wheels, which is approximately ± 210 radians per second under the power constraints of 36 Volts. For low angular velocities, the controller parameters cause the reaction wheels to create a high pitch noise, but the noise is gone once the wheels gain more speed. Time constraints limited further tuning to remove this annoyance.

3.2.2.4 Motor Problems. Once the gyro was properly integrated into the SimSAT architecture, the average run time for SimSAT increased from one minute to over three minutes. The increased run time allowed a new problem with the reaction wheels to surface. Randomly one or more wheels would suddenly lose control. The angular velocity for the reaction wheels is calculated from the change in position of the motor shaft divided by the change in time. The loss of control is due to a problem reading the position of the motor. An example of this problem can be seen in Figure 3.10. The top plot shows just how large of a blow up occurs, while the bottom plot is meant to show just how well the reaction wheel was able to track the commanded value prior to the problem.

Further exploration showed that the error occurred when the onboard computer was unable to process a new position query, so it reported the previous position. Then, the data queue would have two positions to report the following time step. The Simulink®️ code did not know what to do with the extra position data, so it combined them into one position data set. This caused an unrealistically large angular velocity, but the problem was partially solved by creating a filter to only process the most recent position data set whenever two sets are available.

This solution while allowing SimSAT's reaction wheels to recover from the data flow problem is not perfect. The reaction wheels will now have momentary spikes due to the fact that one data packet will be lost on occasion. This causes the reported angular velocity to suddenly drop to zero then spike in the opposite direction before regaining proper tracking. It would be possible to filter out some of these spikes if they occur at a point where the
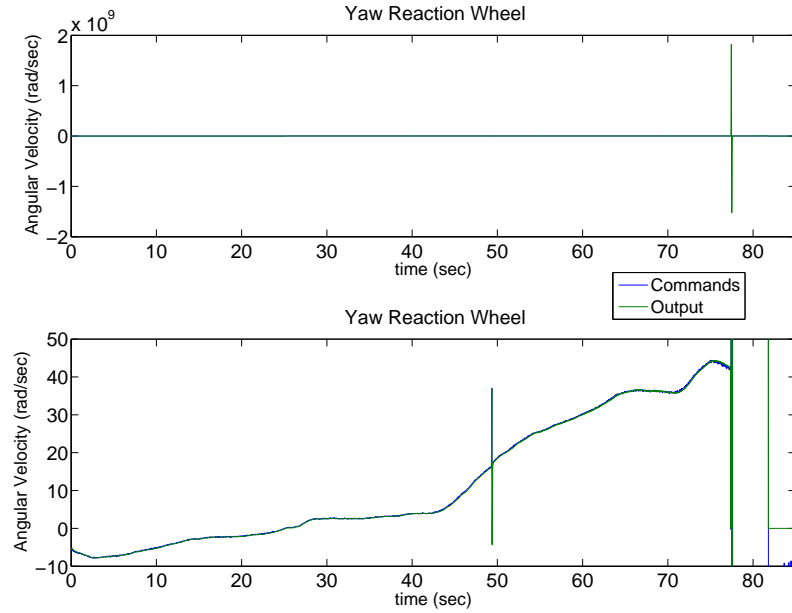
Figure 3.10   Reaction Wheel Velocity Reporting Problem

drop to zero exceeds the torque constraint put on the reaction wheels by the saturation within the control law of SimSAT. It was decided that this filter was not necessary and therefore not implemented due to limited time. In addition, it was easier to remove these spikes after the data was collected from SimSAT and before it was processed in Matlab®.

IV. Results and Analysis

4.1 Testing Procedures

The test procedure for this thesis is relatively simple. Using a position command for the yaw axis as seen in Figure 4.1, SimSAT was commanded for approximately five minutes where the reaction wheel rates and SimSAT's angular rates were recorded. The steps found in Appendix A are a step by step guide for performing a single run.
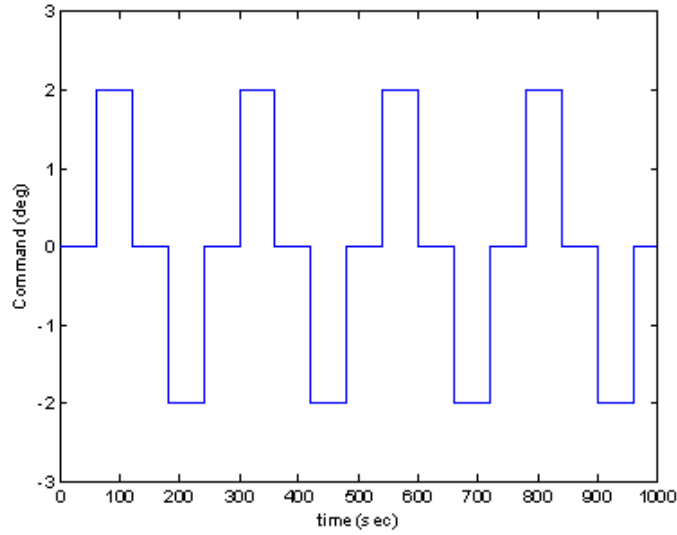

Figure 4.1   Yaw Position Command

In this thesis, several runs for each test setup were done in order to properly estimate the MOI for each test. First, a test was developed to estimate the baseline MOI of SimSAT. This baseline was SimSAT without any water in its tanks. Subsequent tests were done with varying amounts of water in the tanks in order to properly characterize the fuel estimation process.

The amount of water used for each test was measured using a $250mL$ graduated beaker. The water was deionized water produced at AFIT. Distilled water would have been used, but none was readily available, and this water was pure enough to have an approximate density of $1kg/L$. The amount of water in each tank was always the same in

order to preserve SimSAT's balance on the air pedestal. The number of runs for each test configuration can be seen in Table 4.1.

Table 4.1   Test Distribution

Test	Mass of Water	Number of Test Runs
Baseline	0 kg	20
1	1 kg	20
2	2 kg	20
3	3 kg	20

4.2   SimSAT Data Preparation

From these test runs, the reaction wheels' angular velocities and SimSAT's body angular rates were saved for later processing. For continuity, the first 50 seconds are removed and anything after 325 seconds is removed. The first 50 seconds was removed in order to give SimSAT a chance to initialize and remove any initial movement prior to beginning the two degree maneuvers. The data after 325 seconds was removed because the yaw reaction wheel began to become saturated due to the disturbance torque. For some data sets, more data had to be removed from the ends due to the timing of the gyro freeze problem discussed earlier.

4.3   MOI and Fuel Estimation Obstacles

The basic estimation processes were described at the end of Chapter 2, but a few problems need to be addressed before the MOI and fuel estimation processes are ready to be used. Chapter 3 discussed several peculiarities of the SimSAT setup that could have a negative effect on both estimation processes. In addition, a working model of SimSAT is still required for the MOI estimation process.

4.3.1   SimSAT Data.   The problems with the data from SimSAT are the gyro freeze and the wheel speed spike that are described in Chapter 3. Both of these problems causes the MOI to be over estimated. In order to keep the gyro freeze from causing an undesired affect on the estimated MOI, the spots where the gyro data froze were removed

from the calculation of the cost for fmincon along with the next second of data. The code used for this filter can be found in Appendix C.4. In order to keep the spikes in the reaction wheel data from causing problems, a filter for the reaction wheel data is used to remove the spikes in the wheel speed and replace it with a linear interpolation between the two good data points to either side of the spike. The code used for this purpose can be found in Appendix C.3. The peculiarities inherent in the data needed from SimSAT are now accounted for and no loner have negative affects on the MOI estimation process. The final item required for the MOI estimation process to be complete is a working model of SimSAT.

4.3.2 SimSAT Model. As stated in Chapter 2 previous students created models of SimSAT that attempted to simulate SimSAT's Euler angles over time. Once the SimSAT hardware was functioning it became apparent that their models would no longer work with the SimSAT hardware. The gyro freeze does not allow SimSAT to properly track the Euler angles over time. Without reliable data from SimSAT to match the SimSAT model to the estimation process will fail. Since the past models of SimSAT are no longer useful in modelling SimSAT's behavior, a new model that does not model the Euler angles needed to be created.

Originally, we attempted to create a three axis model of SimSAT in order to describe the baseline motion of SimSAT, but this model was unsuccessful. The difficulty of accurately modelling all of the disturbance torques and the limited time remaining once the SimSAT hardware was working properly made the three axis model an unrealistic objective. Some of the disturbance torques for the three axis model that were considered are:

- Onboard Computer Fan Torque,

- Reaction Wheel Misalignment,

- Center of Gravity (COG) Misalignment (not in center of spherical air bearing), and

- Air Currents.

When the three axis model failed, a model of SimSAT about the yaw axis only was created. The model takes the yaw reaction wheel rate data from SimSAT and models the yaw axis dynamics. This is an accurate model of SimSAT because the controller was designed to keep the angular velocity of SimSAT about the roll and pitch axes zero. The pitch and roll reaction wheels were only used to counter disturbance torques about their respective axes. SimSAT's dynamics can then be modelled as motion about the yaw axis only. This leaves only two model parameters to be estimated for the baseline model of SimSAT: the MOI of SimSAT's yaw axis and the constant disturbance torque about the yaw axis. With this model and the use of fmincon optimization a baseline MOI can be estimated. The baseline MOI estimation process requires the angular rate of the yaw reaction wheel and the yaw body angular rate of SimSAT to estimate the yaw MOI and the constant disturbance torque. An example of the data from SimSAT that is used as inputs into the SimSAT model for fmincon is in Figure 4.2.
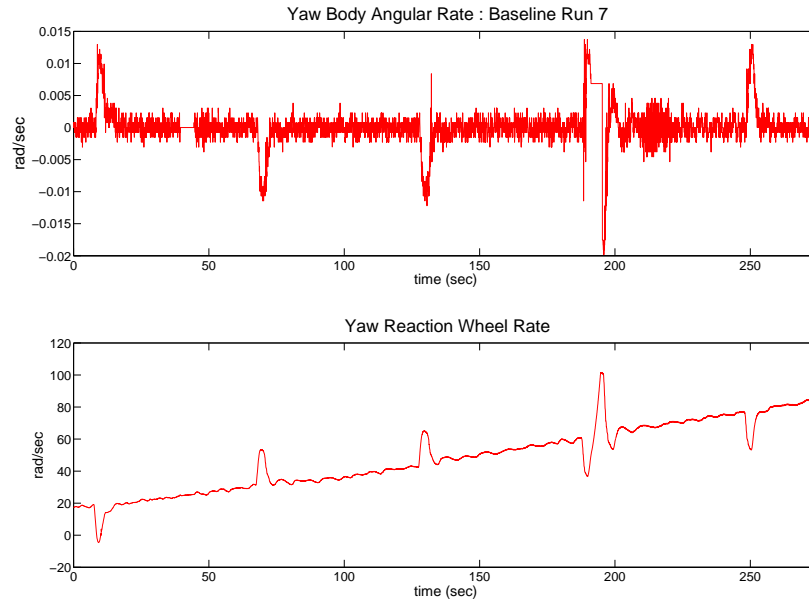

Figure 4.2   Input Data from SimSAT for Model

Next, the model was expanded to include the fuel slosh dynamics to incorporate fuel estimation. This new model integrates the fuel slosh dynamics with the Euler dynamics to describe SimSAT's motion about the yaw axis with water in its tanks. This model

would have required the baseline yaw MOI of SimSAT along with the pendulum length, the bob mass, and the stationary mass from Figure 2.1 and Table 2.1. This model involved integrating the Euler equations of motion and the equations of motion of the three dimensional pendulum and allowing fmincon to find the percent fill of the tanks that provided the least amount of error between the measured and the modelled angular rates. This technique proved to be infeasible because the combination of the dynamics along with the inner workings of fmincon would cause errors involving the algebraic loop incorporated by the interdependence of the 3D pendulum and SimSAT's dynamics. This problem was partially investigated, but due to time constraints it was not solved. Additionally, visual inspection of the water during the test runs showed little to no evidence of sloshing, so it was believed that SimSAT's motion could be modelled accurately enough without the fuel slosh dynamics included. In the article by Keith Schlee, the water was excited at frequencies from 0.5 to 3 hertz [20]. In this thesis, the water in the tanks is excited at frequencies several orders of magnitude less. In addition, the constraint forces measured in the article by Keith Schlee are sinusoidal forces described by accelerations with amplitudes greater than 80 g's, which has a much larger magnitude than that experienced by the water in SimSAT [20]. The differences between the excitation frequencies and the constraint forces explain why the fuel slosh dynamics can be removed without compromising the accuracy of the estimations of the remaining fuel too much.

Since the SimSAT model that included the fuel slosh dynamics was not working, the model originally used to describe SimSAT's baseline motion was used and the water was assumed to be a stationary mass. The Simulink® diagrams in Appendix C model SimSAT's dynamics for all test configurations, and the only variables that the model is dependent on are MOI and disturbance torque about the yaw axis. The model of SimSAT originates from Equation 2.3. Now the MOI estimation processes is complete. Figure 4.3 is an example of the model results after fmincon optimization. The simulated rates are less than the actual rates because the MOI of SimSAT was overestimated due to inaccuracies incorporated into the model by assuming a constant disturbance torque about the yaw axis. The inaccuracy of the model is apparent in the wavy nature of the error plot as it fluctuates away and back to zero. Figure 4.4 is an example of the model results after

fmincon optimization for a good test run. As expected, the error between the actual yaw rate and the simulated yaw rate appears to be noise centered about zero.
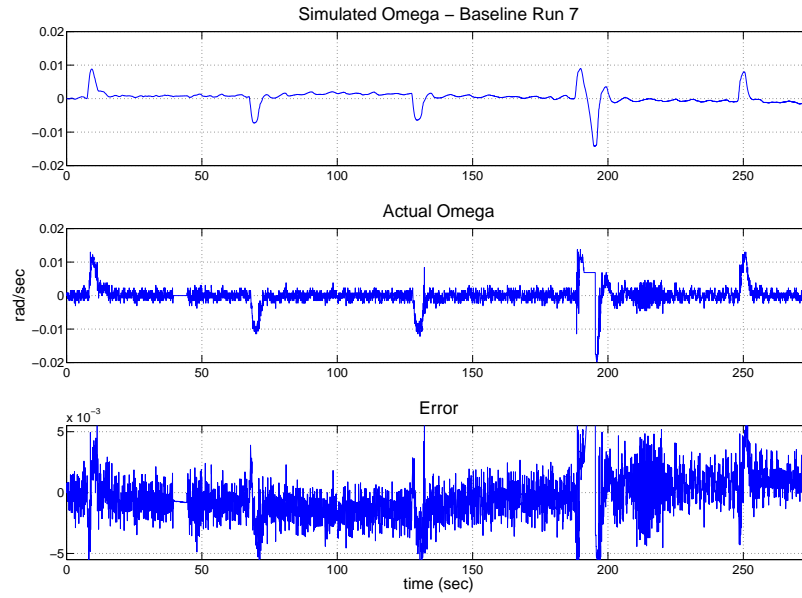

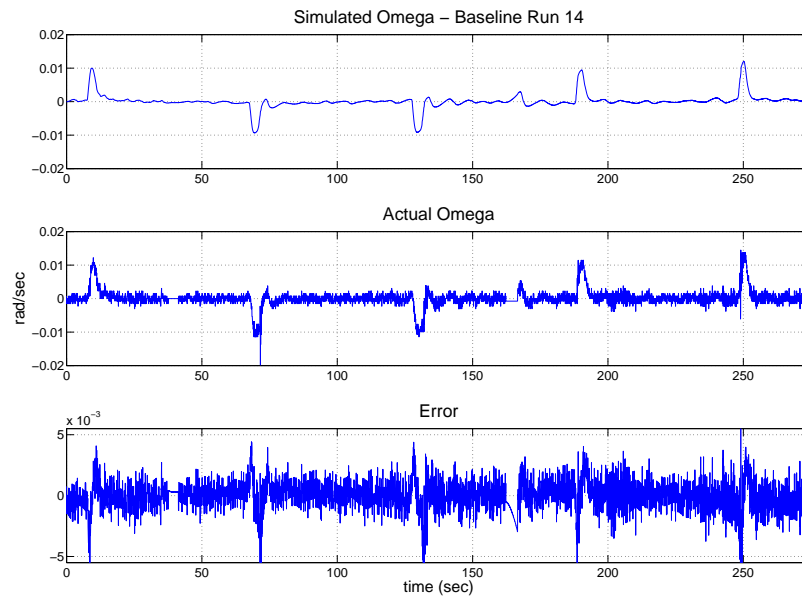Figure 4.3   Model Results - Bad Test

Figure 4.4   Model Results - Good Test

4.4 MOI Estimation

MOI estimation was performed for four configurations of SimSAT where the only difference between them was the amount of water in the attached tanks. The mean and standard deviation of the results of the MOI estimation process for all test runs for the three cost functions are in Table 4.2. As the cost function begins to weight the error vector more at points where SimSAT was moving more than 0.005 rad/sec the mean and standard deviation of the results decrease. The ability of Cost Function 2 and 3 to produce estimates with lower standard deviations suggests that it is capable of producing better results than Cost Function 1.

Table 4.2   Cost Function Effect upon MOI Estimations

	Baseline		Test 1		Test 2		Test 3	
Cost Function	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
1	69.1388	3.3118	70.1398	3.9544	71.7185	1.9947	73.5197	4.2920
2	63.6233	1.4914	64.9066	1.5306	66.5475	1.6446	67.8706	2.3093
3	62.0353	1.6808	63.1943	1.7026	64.9766	1.9496	66.0113	1.6530

The values in Table 4.2 are the results of the MOI estimation technique without any outlying data removed. Once the outliers have been removed with the Grubbs test as discussed in , the results appear as they do in Table 4.3. After Grubbs test has been used to identify and remove the outliers, the results have smaller standard deviations as expected. The decrease in standard deviation means that the confidence in the corresponding estimate has increased. Throughout the processes of eliminating outliers for the data sets from different cost functions a pattern emerged. As the cost function increased the weighting on the error vector when SimSAT was moving, the number of outliers found in the corresponding MOI data sets decreased.

The decrease in outlier data points in data sets from the same test runs suggests that Cost Functions 2 and 3 have a problem identifying which test runs the model of SimSAT can not accurately simulate. Incorporating the MOI estimations from these test runs into the MOI data sets for Cost Functions 2 and 3 cause an unnecessary bias in the resulting means and standard deviations. The results in Table 4.4 use the MOI data sets

Table 4.3   Cost Function Effect upon MOI Estimations
with Outlier Removal

	Baseline		Test 1		Test 2		Test 3	
Cost Function	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
1	68.2993	1.5861	69.5684	1.7059	71.2102	1.1614	71.3107	0.9224
2	63.6233	1.4914	64.7464	1.3896	66.3553	1.0233	66.9312	1.0590
3	62.2202	1.5034	63.1943	1.7026	65.2167	1.6717	65.8471	1.5216

from Cost Function 1 in order to identify which test runs the SimSAT model is unable to accurately simulate. The results for Cost Function 2 and 3 are then the mean and standard deviation of their corresponding MOI data sets without the MOI data points from the test runs identified as outliers by Grubbs test for Cost Function 1. This technique causes the standard deviations to decrease in some cases and to increase in others, so there is no clear affect on the confidence of the estimates. The estimations for the mass of water need to be calculated from the MOI estimations to notice any effects eliminating test runs based on the outliers in Cost Function 1 had on the accuracy of either the MOI or water estimation processes.

Table 4.4   MOI Estimates with Outlier Identification
with Cost Function 1

	Baseline		Test 1		Test 2		Test 3	
Cost Function	Mean	Stdev	Mean	Stdev	Mean	Stdev	Mean	Stdev
1	68.2993	1.5861	69.5684	1.7059	71.2102	1.1614	71.3107	0.9224
2	63.4506	1.4732	64.8712	1.3157	66.1763	1.2499	66.8880	1.1302
3	62.0660	1.7019	63.4565	1.5799	64.6342	1.7317	65.5478	1.3583

Tables 4.3 and 4.4 each have three MOI estimations for each test configuration. The MOI estimates have standard deviations of the same magnitude. It is impossible to statistically decide which MOI estimation is more accurate without a known true value to compare them to. Without a known true value, the theory behind the MOI estimation process along with the meaning of the cost functions need to be considered. The MOI estimation processes relies on the ratio between the input torque and the change in the

body's angular rate to calculate the experimental MOI. During the test runs the MOI estimation processes is occurring when SimSAT has a changing angular rate about the yaw axis when the yaw reaction wheel produces a torque to change the attitude of SimSAT. When SimSAT is attempting to maintain the current attitude the MOI estimation processes does not apply because the angular rate of SimSAT can be approximated as a constant zero radians per second. Cost Function 1 weights all of the data equally in the MOI estimation process, and as just discussed, the data points where SimSAT is not moving do not provide reliable data to the MOI estimation. The results from Cost Function 1 should be the least trusted. The MOI estimation process gets more accurate as the weighting on the data where SimSAT is moving increases. Therefore, based on the principles behind the MOI estimation processes Cost Function 3 should have results that are the least affected by noise and inaccuracies in the SimSAT model, and should therefore be given more credit than the results from the other two cost functions.

4.5   Fuel Mass Estimation

From the estimated MOI data for SimSAT outlined in Section 4.4 the water in the three different test configurations can be estimated with the processes outlined in Section 2.4.2. Using the MOI data from Table 4.3, the estimated mass of water can be seen in Table 4.5.

Table 4.5   Mass of Water Estimations

	Test 1 (1 kg)		Test 2 (2 kg)		Test 3 (3 kg)	
Cost Function	Mean	Stdev	Mean	Stdev	Mean	Stdev
1	1.4298	3.7273	3.2720	3.1108	3.3846	2.8402
2	1.2658	3.2619	3.0716	2.8471	3.7167	2.8875
3	1.0982	3.6300	3.3680	3.5950	4.0738	3.4250

The results in Table 4.5 show several inconsistencies of the fuel estimation process. For Test 1 the estimated mass has a percent error of 42.98% for Cost Function 1 but only 9.82% for Cost Function 3. Earlier, Cost Function 3 was predicted to provide the most accurate results, so the decrease in the percent error is expected. However, for Test 3 the

estimated mass has a percent error of 12.82% for Cost Function 1 and 35.79% for Cost Function 3. The affects of the cost functions is erratic in the results in Table 4.5, so no conclusions about the affection of the cost functions can be made. The standard deviation of the results on the other hand are much too large for the fuel estimation technique to ever be useful in real applications.

The percent error in the mass of water estimation is in Table 4.6. For Test 1 the progression of the cost functions decreased the percent error of the mass of water estimation. For Test 2 the percent error remain approximately the same, and for Test 3 the percent error increased. Using the different cost functions has an unpredictable affect on the water estimation processes when Grubbs test is used to identify outliers for each individual MOI estimation data sets. One one estimate from all of Table 4.6 has a percent error low enough to be considered an accurate estimate.

Table 4.6   Percent Error of Mass of Water Estimations

Cost Function	Test 1	Test 2	Test 3
1	42.98	63.6	12.82
2	26.58	53.58	23.89
3	9.82	68.4	35.79

Using the MOI data from Table 4.4 that was filtered using Grubbs test on Cost Function 1 to identify outlier test runs for all cost functions, the estimated masses of water is in Table 4.7. The percent error for the estimated masses of water is in Table 4.8.

Table 4.7   Mass of Water Estimations with Outlier Identification with Cost Function 1

	Test 1 (1 kg)		Test 2 (2 kg)		Test 3 (3 kg)	
Cost Function	Mean	Stdev	Mean	Stdev	Mean	Stdev
1	1.4298	3.7273	3.2720	3.1108	3.3846	2.8402
2	1.6001	3.1577	3.0644	3.0832	3.8616	2.9477
3	1.5664	3.7157	2.8880	3.8875	3.9114	3.4648

Table 4.8   Percent Error for Mass of Water Estimations

Cost Function	Test 1	Test 2	Test 3 (kg)
1	42.98	63.6	12.82
2	60.01	53.22	43.08
3	56.64	44.4	30.38

The percent errors in Table 4.8 are too large to consider the water estimations accurate just as the results in Table 4.6. Except for Test 1, removing the outliers specified by Cost Function 1 for the rest of the cost functions did decreased the percent error when compared to Table 4.6. This suggests that using Cost Function 1 to identify test runs that the SimSAT model can not accurately simulate is justified.

The effect of the cost functions on the error of the estimates in Table 4.8 are the same as seen earlier in Table 4.6. However, studying the masses of water estimated using Cost Function 3 in Table 4.7 shows that the difference between the estimated masses of water comes to approximately the difference in the actual mass of the tests. The differences between the estimated masses for the tests are in Table 4.9.

Table 4.9   Mass of Water Difference between Tests Estimated and Actual

	Test2 - Test 1		Test 3 - Test 2		Test 3 - Test 1	
Cost Function	Estimated	Actual	Estimated	Actual	Estimated	Actual
1	1.8422	1	0.1126	1	1.9548	2
2	1.4643	1	0.7972	1	2.2615	2
3	1.3216	1	1.0234	1	2.3450	2

The results in Table 4.9 support that the estimation processes used in this thesis may still be able to provide more accurate estimates for the differences in fuel between two configurations. The increased accuracy of the estimations of the differences of mass between the tests suggests that the baseline MOI was not accurately estimated. However, this argument is pointless until the standard deviation of the estimations from the MOI estimation process are decreased. Without a higher degree of confidence in the estimated

MOI values needed for the fuel estimation process, the fuel estimation process will never be useful in real world applications.

V. Conclusions and Recommendations

5.1 Conclusions

From the results shown, it can be concluded that the fuel estimation process studied in this thesis is not accurate enough in its present state to provide an increase in the current abilities of fuel estimation due to other instruments and techniques with much higher accuracies and smaller uncertainties. This thesis was unable to completely characterize the estimation technique due to limited time and the poor initial condition of the hardware and software. Further testing to characterize the process needs to be done. While the initial results do not provide any improvement over the current fuel estimation processes in use, the development of this process should not be stopped. This thesis has shown that the theory behind the process is viable, but the accuracy of its application is not what it needs to be in order to implement the process in real world situations. There are many ways to make this technique more accurate. The increase in the fidelity of the disturbance torque models along with hardware upgrades would have the largest effect on the accuracy of the estimation process. In addition, a theoretical simulation of the technique would be able to ascertain the peak performance possible for the fuel estimation process.

5.1.1 Sources of Error.   There are a number of error sources throughout the estimation process. The yaw reaction wheel is not perfectly aligned with the yaw axis. The air currents in the lab are unpredictable and constantly changing due to the building's climate control. There are moments created by the COG of SimSAT not being in the center of the air bearing. SimSAT is a flexible structure. The list of small but persistent disturbance forces are not the dominant forces affecting SimSAT, but they are constantly there and build up over time.

The main reason why the model of SimSAT can not accurately simulate its motion is that the disturbance torques about the yaw axis are not accurately modelled as a constant value. A previous student installed cardboard deflectors to direct the flow of air away from SimSAT in order to lessen this affect. The undesired affect is still there though lessened, and it is the dominant disturbance torque about the yaw axis. Variations from the constant value assumption causes the MOI to be overestimated. Assuming this problem

affects random runs it should theoretically have no affect on the fuel estimation since it would happen to both the baseline and the test estimations of MOI. In this thesis, the variations on the disturbance torque does have an effect due to the limited number of runs performed for each test, but the effect it has can not be predicted since it could either decrease or increase the estimated fuel mass depending if it affected the baseline or the test runs.

Another source of error for estimating the MOI of SimSAT can be found in the misalignment of the yaw reaction wheel. Whenever the yaw reaction wheel is used, a corresponding torque can be seen about the roll axis. This means that all MOI's estimated in this thesis should be greater than the true value due to the fact that in reality less torque is experienced about the yaw axis than the estimation model believes. This affects all of the MOI estimations equally, so it will have a negligible affect on the fuel estimation.

5.1.2 Application. In order to apply this technique in a real world situation, an accurate attitude dynamics model of the satellite in question would have to be created. There are two major obstacles to this technique being used in real world applications. Modelling the disturbance torques experienced by satellites such as solar pressure and micro-meteorites is one difficulty, and modelling fuel slosh is another. Once the contribution of disturbance torques and fuel slosh to the attitude dynamics is known, the MOI of a satellite can be found to a much higher degree of accuracy.

Small changes in the orientation of SimSAT were used in order to simulate the on-orbit behavior of an average satellite that makes small corrections throughout its lifespan. The use of larger orientation maneuvers and larger control torques should increase the accuracy of this process. The disturbance torques that are not accurately modelled would be smaller relative to the total torque experienced by the satellite for a longer period of time causing a theoretical increase in the accuracy of the fuel estimation process. The validity of this hypothesis was not confirmed using the SimSAT hardware due to time constraints, but could be validated with more testing.

5.2   Recommendations

A ducting system to block or redirect the air vents inside of the SimSAT lab should be installed in order to remove the main source for air currents. This should decrease the second largest contributor to the disturbance torques and remove the largest source of variability about the yaw axis. In addition, upgrading the motors for the reaction wheels may be able to increase the fidelity of any dynamics model. Preferably, the new motor should be able to accurately measure its angular rate. The current motor's abilities to measure angular rates is very poor, which is why the change in shaft position must be used.

A student should be given a thesis objective of creating a three axis dynamical model of SimSAT along with the disturbance torques that SimSAT experiences. The largest problem will lie in characterizing the COG misalignment torque. In order to calculate the torque the total mass of SimSAT must be known along with the exact position of the COG of SimSAT with respect to the center of the spherical bearing. With this accurate model, further study of this process will most likely show an increase in the accuracy of the estimated fuel mass.

Appendix A. Test Procedures

A.1 Start Up

In order to start up SimSAT and use the programs for this thesis, follow the following list of steps. This list will walk you through step by step opening the experiment file for SimSAT allowing the user to control SimSAT's yaw euler angle.

1. Start up the SimSAT desktop, not the one hooked up to the network. The profile name and the password for the computer is "user".

2. Double click the dSPACE ControlDesk shortcut icon on the desktop seen in Figure A.1.

Figure A.1   dSpace Icon on Desktop

3. Once the program has opened it should look like Figure A.2.
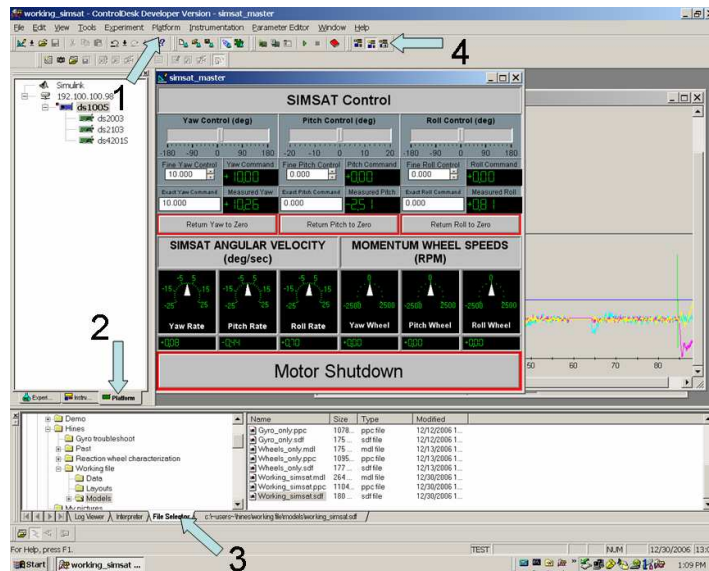
Figure A.2   Initial Control Desk Window

4. After the program is open, load the experiment file that corresponds to the programs you would like to run on SimSAT.

5. Now is the time to turn on SimSAT.

   (a) Flip the three power switches in the order 12, 24, 36. Their location can be seen in Figure A.3. They are labelled to insure that they are turned on in the correct order.
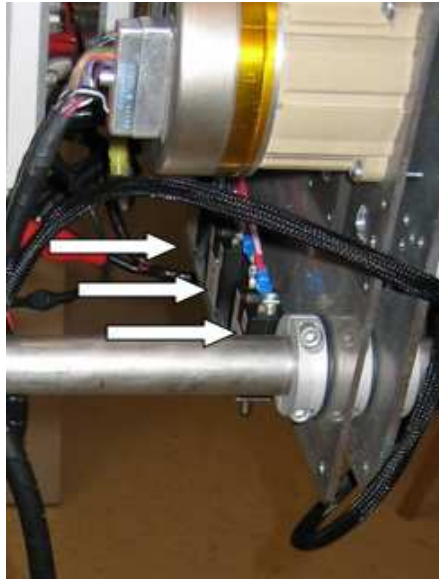
Figure A.3   SimSAT Power Switches

   (b) Once the switches are flipped an initialization process is begun. When it is done you will hear two beeps.

6. In Control Desk you need to connect the computer and SimSAT. To do this go into menu Platform/Change Connection/Network seen in Figure A.2 pointed out by arrow number 1.

7. Select Wireless Connection and use the IP Address of 192.100.100.98 to connect to SimSAT.

8. Click on the Platform tab in the Control Desk Navigator seen in Figure A.2 pointed out by arrow number 2.

9. Click the file selector tab on the bottom of the Control Desk window highlighted by arrow number 3 in Figure A.2.

10. Find the *.sdf file associated with your experiment, and click and drag the file onto the DS1005 seen in the Platform window displayed after clicking the Platform tab. This begins the process of downloading the required files to SimSAT.

11. The experiment that you loaded will begin immediately, so be prepared. Once the play icon turns green click on the icon to go into animation mode pointed to by arrow number 4 in Figure A.2. This mode will allow the layouts to display current telemetry data from SimSAT.

12. The data from the reaction wheels and the gyro are automatically collected, and are saved under the file name you chose once the experiment is over.

A.2 Shut Down

Now that you are done with SimSAT this process walks you through shutting SimSAT down.

1. Stop the running experiment. You do this by switching to a mode other than animate then click the red square to stop the experiment.

2. Once the experiment has stopped, change the connection to Bus rather than Network using the menu Platform/Change Connection/Network.

3. Flip off the power switches in reverse order (36,24,12).

4. Secure SimSAT so that it can not rotate.

Appendix B. Data from SimSAT

This chapter has 6 out of 20 plots for the four different tests. SimSAT has empty tanks for the baseline test. Three kilograms of water are in the tanks for Test 1 and Test 3. Test 2 has one kilogram of water in the tanks. The positive linear trend of the yaw wheel's angular velocity for all of the results is what led to the modelling of the disturbance torque about the yaw axis as a constant value. Run 1 and 4 in Figure B.1 both show the data spikes in the reaction wheel angular velocity discussed in Chapter III. All of the plots display the data freeze of the gyro data at approximately two minute intervals.
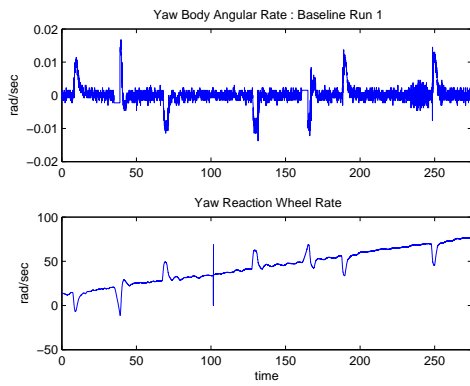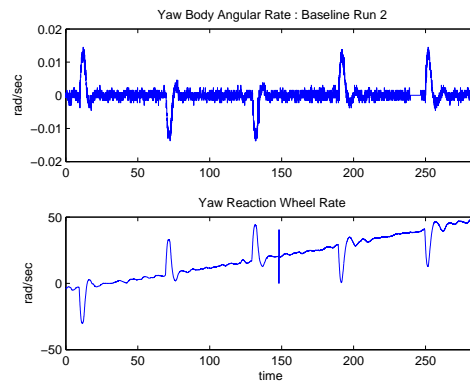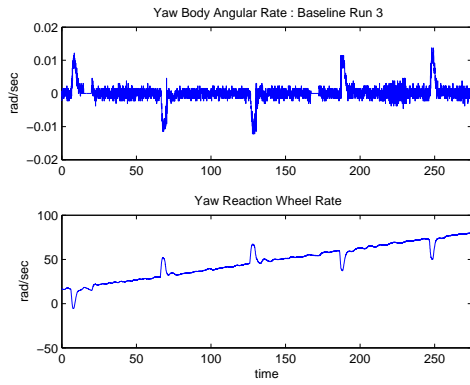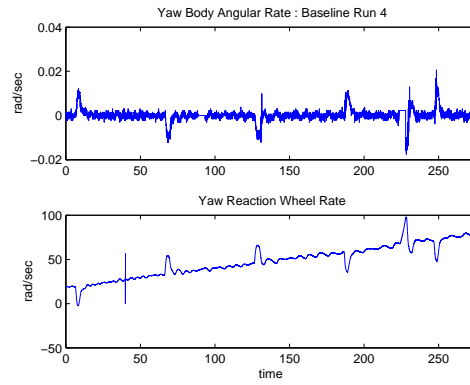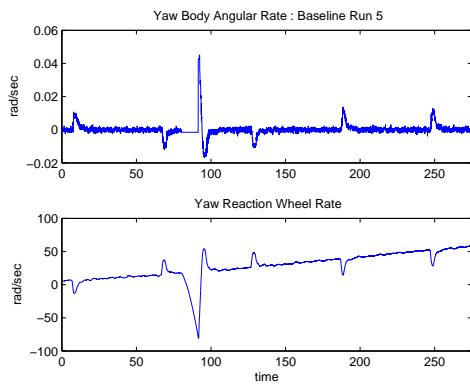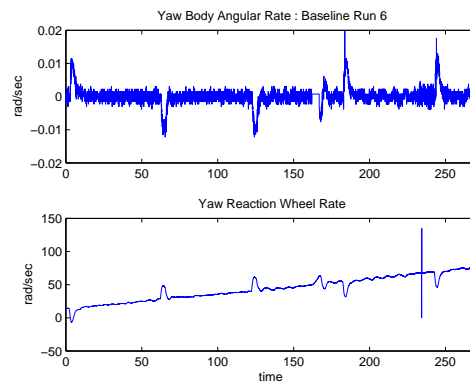
Figure B.1   Six Data Plots from SimSAT for Baseline
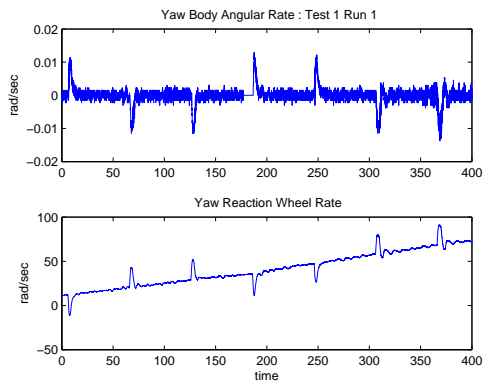
Run 1
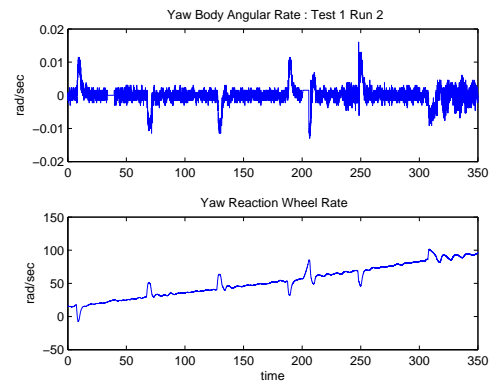
Run 2
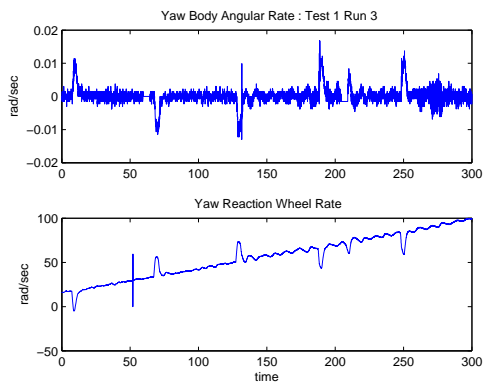
Run 3
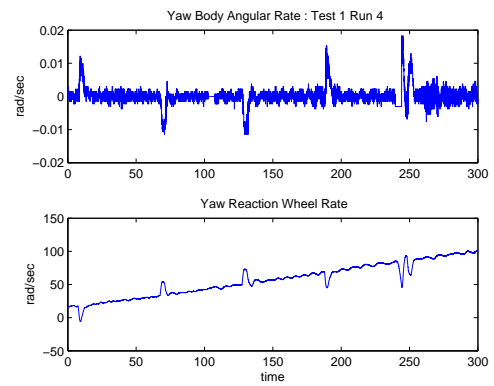
Run 4
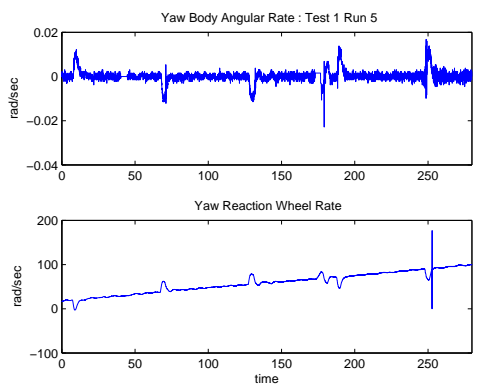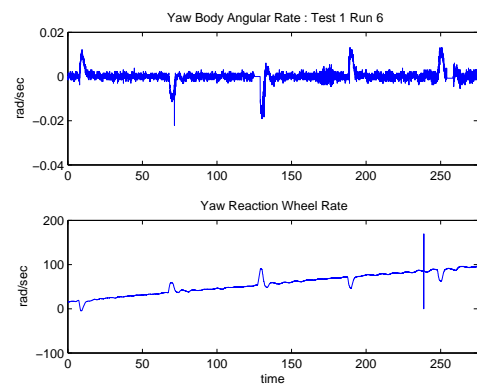
Run 5

Run 6

Figure B.2   Six Data Plots from SimSAT for Test1
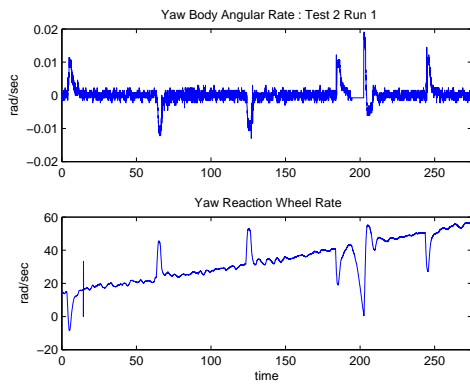
Run 1
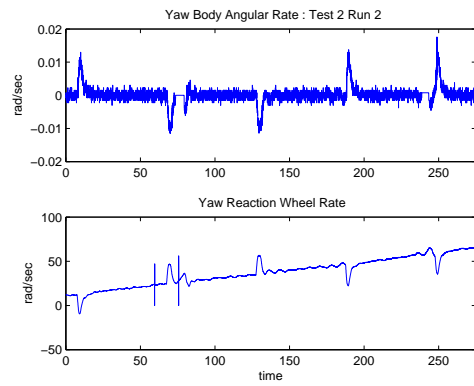
Run 2

Run 3
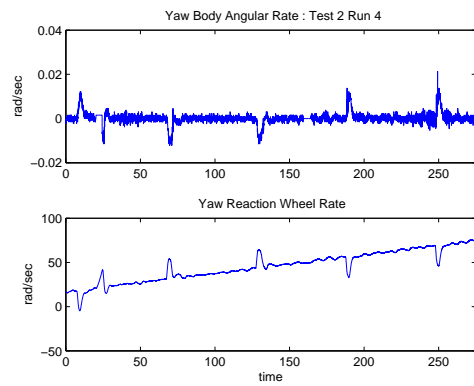
Run 4
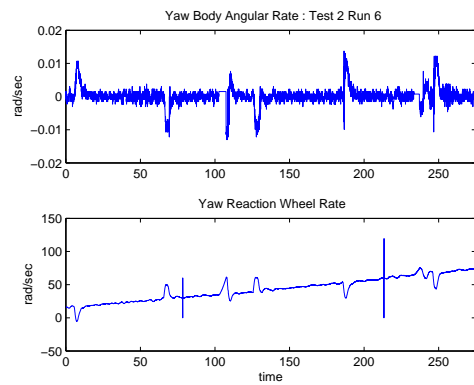
Run 5

Run 6

Figure B.3   Six Data Plots from SimSAT for Test2
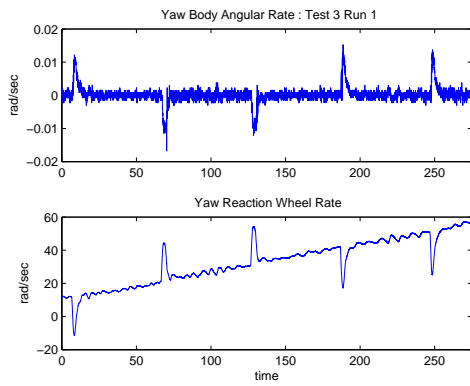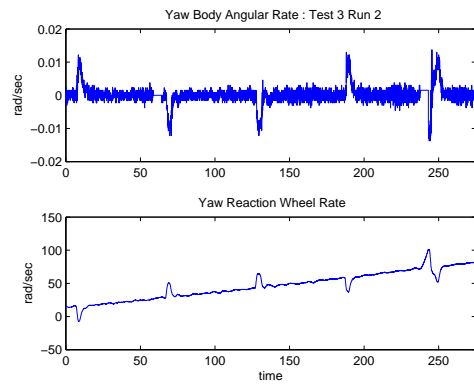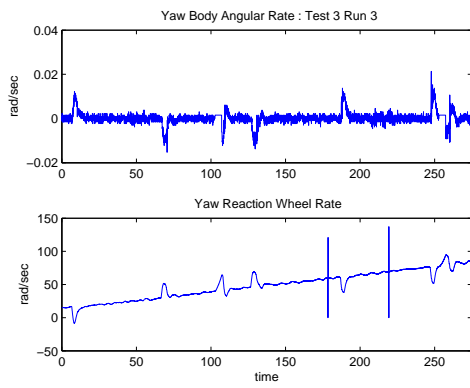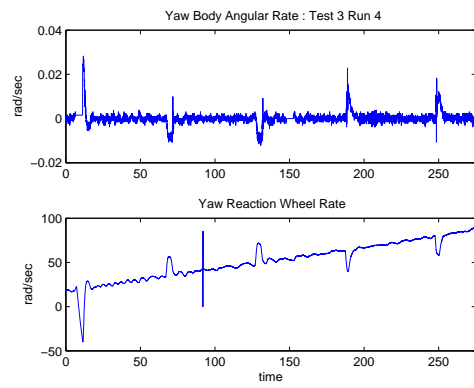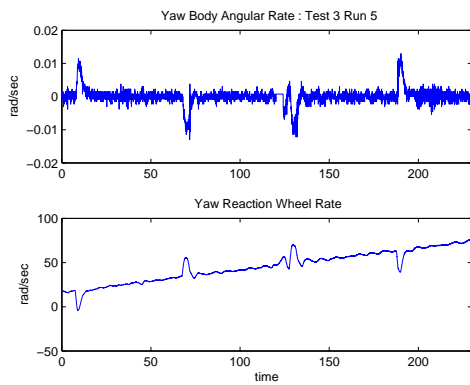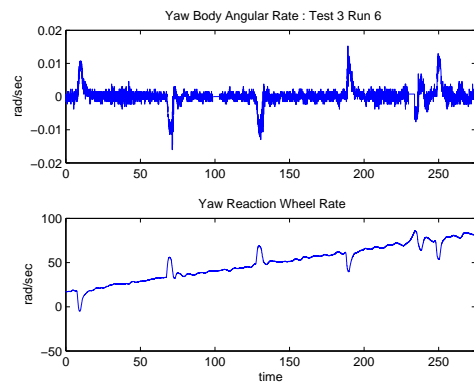
Figure B.4   Six Data Plots from SimSAT for Test3

Appendix C. Data Processing Code

C.1 MOI Estimation Process

Listing C.1   Model matching main code

```
   close all
   clear all
   clc

 5 global Wheel_0 omega_0 theta_0 I_rw I_ss I_ss_inv M Omega_RW Omega_SS ...
       gamma

   %% Program Description
   % This program uses the inputs from the Yaw Characterization data batch ...
       in
   % order to approximate the inertia about the Yaw axis and the disturbance
10 % torque due to air currents about the yaw axis.

   %% Initialize Variables
   I_rw=(1.955E-2+2.12E-4);                    % reaction wheel inertia
   I_ss=[5 0 0;0 64.248 0;0 0 64.248];     % baseline for SimSAT MOI (not ...
       used)
15 I_ss_inv=I_ss^-1;

   %% Begin Search

   x0=[65.4472,0.0067]; % Estimate Inertia and Moment about Yaw
20
   options=optimset('Display','Iter','TolX',1E-222,'TolFun',1E-222,...
       'TolCon',1E-222,'MaxFunEvals',500,'MaxIter',50);

   lb=[0 0];          % lower bound of fmincon variables
25 ub=[100 1];        % upper bound of fmincon variables

   count=1;

   for test=1:4
30     x = fmincon(@(x) SS_cost_function(x,test),x0,[],[],[],[],lb,ub,[],...
           options)

       [cost,time,output]=SS_cost_function(x,test);

       figure
35     subplot(3,1,1)
       plot(time,output(:,4:6))
       title(['Simulated Omega - Test ' num2str(test)])
       subplot(3,1,2)
       plot(time,output(:,7:9))
40     title(['Actual Omega - Test ' num2str(test)])
       subplot(3,1,3)
       plot(time,output(:,1:3))
       title(['Error - Test ' num2str(test)])

45     final_answer(count)={x};
```

```
        cost_save(count)=cost;
        count=count+1;
    end

50 %% Statistical Analysis
   temp=length(final_answer);

   for i = 1:temp
        temp2=cell2mat(final_answer(1,i));
55      Inertia(i)=temp2(1,1);
        Torque(i)=temp2(1,2);
        clear temp2
    end

60 Inertia_avg=mean(Inertia);
   Inertia_std=std(Inertia);

   Torque_avg=mean(Torque);
   Torque_std=std(Torque);
```

Listing C.2   Model matching cost function code.

```
   function [cost,t,y]=SS_cost_function(x,test)

   global Wheel_0 omega_0 theta_0 I_rw I_ss I_ss_inv M Omega_RW Omega_SS ...
       gamma

 5 %% Program Description
   % This program works with Model_optimization.m in order to provide a cost
   % function that fmincon attempts to drive to zero.  The two variables ...
       that
   % fmincon can change are the inertia about the yaw axis and the constant
   % disturbance torque about the yaw axis.  Cost is computed by subtracting
10 % the simulated data from the actual data and taking the norm of the ...
       error
   % vector for the yaw axis.

   %% Define fmincon variables
   I_ss(3,3)=x(1);
15 I_ss_inv=I_ss^-1;
   M=[0;0;x(2)];

   %% Get Input Data

20 [Time Omega_b Wheels]=data_input_yaw_characterization(test);


   % Find and filter out yaw wheel data spikes
   [Yaw_fixed]=reaction_wheel_filter(Wheels);
25
   % Remove Roll and Pitch Wheels
   temp=length(Wheels);
   Wheels=[zeros(temp,2) Yaw_fixed];

30 % Find any spots with gyro freeze
   [Data_frozen]=yaw_freeze_locations(Omega_b);
```

```
   theta_0=[0;0;0];               % initial Euler Angles
   gamma=0*pi/180;                % Missalignment of Yaw Reaction Wheel
35 omega_0=[0;0;0];               % initial angular rate of SimSAT
   Wheel_0=Wheels(1,1:3)';        % initial angular rate of Reaction Wheels
   Omega_RW=[Time Wheels];        % simulation input for Reaction Wheels
   Omega_SS=[Time Omega_b];       % simulation input for SimSAT's measured ...
      rates

40 [t,x,y]=sim('SS_empty_tank_model_estimation',[Time(1) Time(length(Time))...
      ],[]);

   if Data_frozen==0
       cost=norm(y(:,3));         % cost if no gyro freeze
   else
45     [a,b]=size(Data_frozen);
       error=[];
       for i=1:a+1
           if i == 1
               error=[error;y(1:Data_frozen(i,1),3)];
50         elseif i<=a
               error=[error;y(Data_frozen(i-1,2)+50:Data_frozen(i,1),3)];
           else
               error=[error;y(Data_frozen(i-1,2)+50:length(y),3)];
           end
55     end
       cost=norm(error);          % cost with gyro freeze removed
   end
```

Listing C.3   Reaction Wheel Spike Filter.

```
   function [Yaw_fixed]=reaction_wheel_filter(Wheels)

   %% Program Description
   % When SimSAT has trouble keeping up reading the position counts from
 5 % the motors a spike occurs, and this program finds that spike and
   % replaces it with a linear approximation between the two good data
   % points.

   %% Program
10
   Yaw=Wheels(:,3);

   temp=length(Yaw);

15 for i = 1:temp-3
       if Yaw(i+1)==0 & Yaw(i)~=0 & Yaw(i+2)~=0
           % this identifies the beginning of a spike
           slope=(Yaw(i+3)-Yaw(i))/(0.05*3);
           Yaw(i+1)=slope*0.05+Yaw(i);
20         Yaw(i+2)=slope*0.1+Yaw(i);
       end
   end

   Yaw_fixed=Yaw;
```

Listing C.4   Gyro Freeze Locater.

```matlab
function [Data_frozen]=yaw_freeze_locations(Omega_b)

%% Program Description
% This program will find the cell indexes that identifiy where the
% gyro data has frozen, so that it can be removed from the data.

temp=length(Omega_b);
range=20;
problem_spots=[];
junk=[];

count=1;
for j=1:temp-range
    if Omega_b(j:j+range,3)==Omega_b(j,3)
        problem_spots(count,1)=j;
        count=count+1;
    end
end

a=isempty(problem_spots);

if a==1
    Data_frozen=0;
else
    % start of problem
    temp1=length(problem_spots);
    count=1;
    for j = 1:(temp1-1)
        if problem_spots(j,1)+1==problem_spots(j+1,1)
            junk(count,1)=j+1;
            count=count+1;
        end
    end
    clear count j
    problem_spots_orig=problem_spots;
    problem_spots(junk,1)=0;
    count=1;
    for j = 1:temp1
        if problem_spots(j,1)==0
            % do nothing
        else
            Data_frozen(count,1)=problem_spots(j,1);
            count=count+1;
        end
    end
    clear count j junk
    junk=[];
    problem_spots=problem_spots_orig;
    temp1=length(problem_spots);
    count=1;
    for j = 1:(temp1-1)
        if problem_spots(j,1)+1==problem_spots(j+1,1)
            junk(count,1)=j;
            count=count+1;
        end
```

```
        end
        problem_spots_orig=problem_spots;
        problem_spots(junk,1)=0;
        clear count j
60      count=1;
        for j = 1:temp1
            if problem_spots(j,1)==0
                % do nothing
            else
65              Data_frozen(count,2)=problem_spots(j,1)+range;
                count=count+1;
            end
        end
    end
```
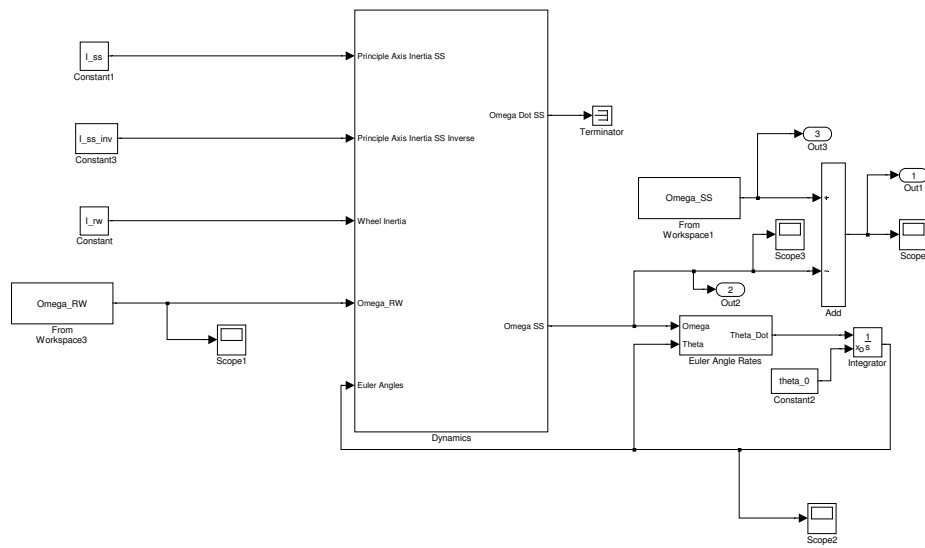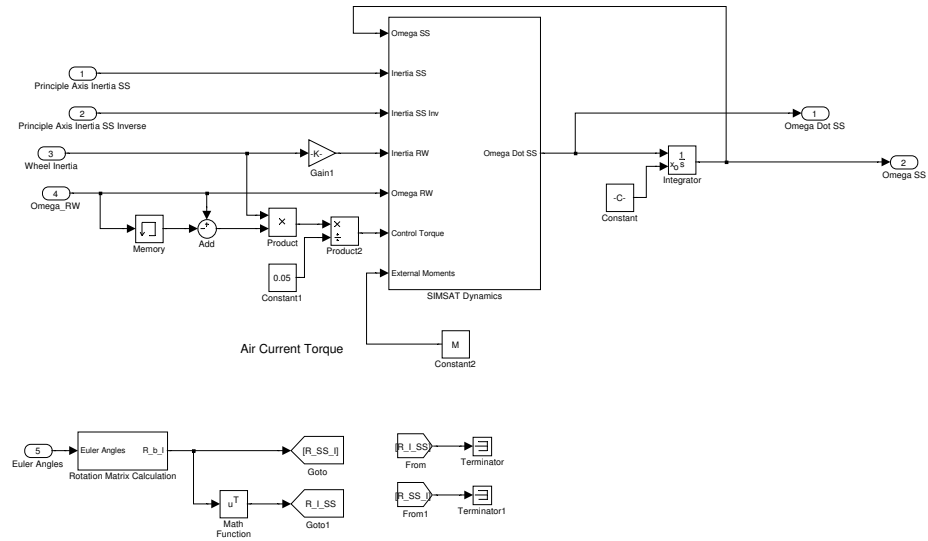

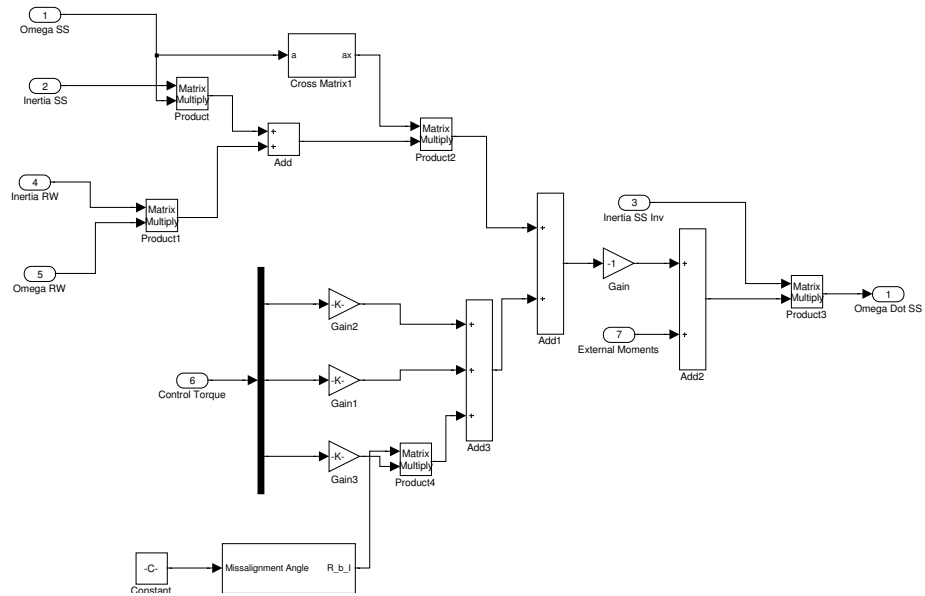Figure C.1   SimSAT Model Top Level

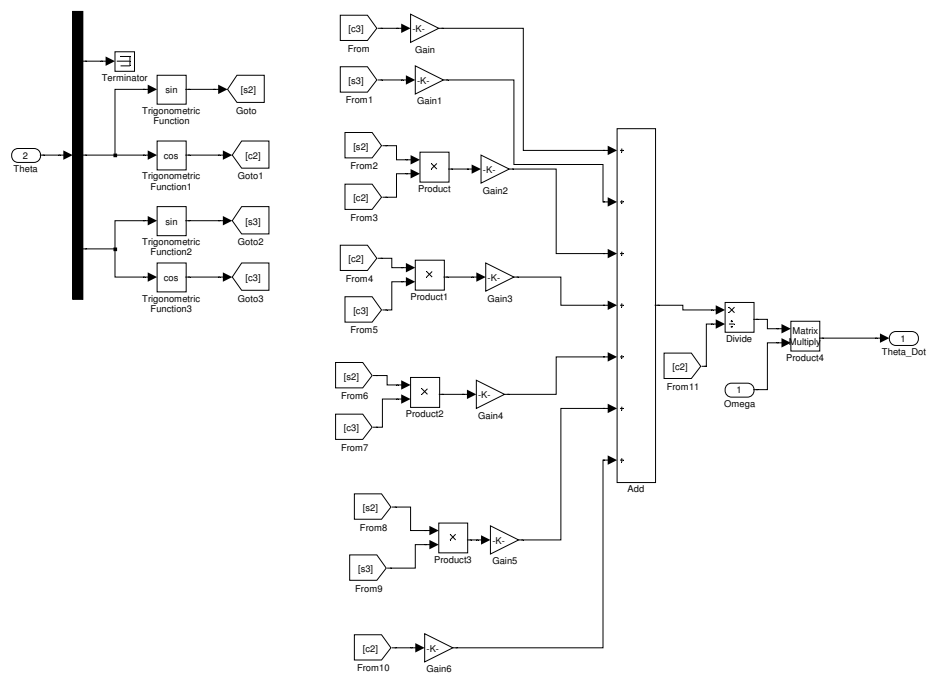Figure C.2   Dynamics Top Level

Figure C.3   Main Dynamics of SimSAT

Figure C.4   SimSAT Euler Rates

Listing C.5   MOI and Water Estimation Analysis.

```
   close all
   clear all
   clc

5  %% Program Desction
   % This program allows systematic comparison of all factors being examined
   % in the estimation process for fuel on SimSAT.

   %% Program
10 cost_function=3;
       % 1 -> cost function 1 (equal weighting)
       % 2 -> cost function 2 (2x weighting for movement)
       % 3 -> cost function 3 (5x weighting for movement)
   test=1;
15     % 1 -> test 1 (20 runs with 3kg water)
       % 2 -> test 2 (20 runs with 1kg water)
       % 3 -> test 3 (20 runs with 2kg water)
   plots_baseline=0;
       % 1 -> plot baseline data progression
20     % 0 -> do not plot baseline data progression
   plots_test=0;
       % 1 -> plot test data progression
       % 0 -> do not plot test data progression
   technique=3;
25     % 1 -> Grubbs technique for all tests and cost functions
       % 2 -> Grubbs technique for cost function 1 used throughout
       % 3 -> Removed all that Grubbs identified for all three cost
             % functions

30 L=74/2*0.0254; % length between ideal COG and tank COG

   % load baseline data
   load Test0
   Inertia_Baseline=cell2mat(Test0(cost_function,1));
35 Error_mean_Baseline=cell2mat(Test0(cost_function,2));
   Error_sum_squares_Baseline=cell2mat(Test0(cost_function,3));
   Sim_data_Baseline=Test0{cost_function,4};
   clear Test0

40 % load test data
   if test==1
       load Test1
       Inertia_Test=cell2mat(Test1(cost_function,1));
       Error_mean_Test=cell2mat(Test1(cost_function,2));
45     Error_sum_squares_Test=cell2mat(Test1(cost_function,3));
       Sim_data_Test=Test1{cost_function,4};
       clear Test1
   elseif test==2
       load Test2
50     Inertia_Test=cell2mat(Test2(cost_function,1));
       Error_mean_Test=cell2mat(Test2(cost_function,2));
       Error_sum_squares_Test=cell2mat(Test2(cost_function,3));
```

```matlab
        Sim_data_Test=Test2{cost_function,4};
        clear Test2
55 elseif test==3
        load Test3
        Inertia_Test=cell2mat(Test3(cost_function,1));
        Error_mean_Test=cell2mat(Test3(cost_function,2));
        Error_sum_squares_Test=cell2mat(Test3(cost_function,3));
60      Sim_data_Test=Test3{cost_function,4};
        clear Test3
    end

    if technique==1
65      % Baseline Calculations
        [Removed]=Grubbs_Test(Inertia_Baseline)
        if plots_baseline==1
            name='Grubbs Test: Baseline MOI';
            data_progression_plot(Inertia_Baseline,Removed,name);
70          clear name
        end
        Inertia_Baseline(Removed)=[];

        Baseline_MOI=mean(Inertia_Baseline);
75      Baseline_STD=std(Inertia_Baseline);

        [Removed]=Grubbs_Test(Inertia_Test)
        if plots_test==1
            name='Grubbs Test: Test MOI';
80          data_progression_plot(Inertia_Test,Removed,name);
            clear name
        end
        Inertia_Test(Removed)=[];

85      Test_MOI=mean(Inertia_Test);
        Test_STD=std(Inertia_Test);

        [m_est,I_fuel]=Inertia_to_mass(Baseline_MOI,Test_MOI,L,1000);

90      fuel_mass_est=m_est;
        fuel_mass_std=(Baseline_STD+Test_STD)/L^2;
    elseif technique==2
        Removed_baseline=[7 20];
        Inertia_Baseline(Removed_baseline)=[];
95      Baseline_MOI=mean(Inertia_Baseline)
        Baseline_STD=std(Inertia_Baseline)

        if test==1
            Removed_test1=[16 1 2 10 17 5];
100         Inertia_Test(Removed_test1)=[];
        elseif test==2
            Removed_test2=[10 9];
            Inertia_Test(Removed_test2)=[];
        elseif test==3
105         Removed_test3=[14 9];
            Inertia_Test(Removed_test3)=[];
        end
```

```
        Test_MOI=mean(Inertia_Test)
110     Test_STD=std(Inertia_Test)

        [m_est,I_fuel]=Inertia_to_mass(Baseline_MOI,Test_MOI,L,1000);

        fuel_mass_est=m_est;
115     fuel_mass_std=(Baseline_STD+Test_STD)/L^2;
    elseif technique==3
        Removed_baseline=[5 7 20];
        Inertia_Baseline(Removed_baseline)=[];
        Baseline_MOI=mean(Inertia_Baseline)
120     Baseline_STD=std(Inertia_Baseline)

        if test==1
            Removed_test1=[1 2 5 10 16 17];
            Inertia_Test(Removed_test1)=[];
125     elseif test==2
            Removed_test2=[9 10];
            Inertia_Test(Removed_test2)=[];
        elseif test==3
            Removed_test3=[9 13 14];
130         Inertia_Test(Removed_test3)=[];
        end

        Test_MOI=mean(Inertia_Test)
        Test_STD=std(Inertia_Test)
135
        [m_est,I_fuel]=Inertia_to_mass(Baseline_MOI,Test_MOI,L,1000);

        fuel_mass_est=m_est;
        fuel_mass_std=(Baseline_STD+Test_STD)/L^2;
140 end

    clear Sim_data_Test Sim_data_Baseline cost_function mean_limit...
        num_std plots_test sum_limit technique test Removed L ...
        Error_mean_Test Error_mean_Baseline Error_sum_squares_Test ...
145     Error_sum_squares_Baseline plots_baseline
```

Listing C.6   Grubbs Test Coding.

```
    function [Removed]=Grubbs_Test(x)
    % close all
    % clear all
    % clc
  5 %
    % x=[100 2 3 1 2 3 2 3 4 2 1 3 2 2 3 4 2 -200];

    %% Grubbs Test
    Removed=[];
 10 x_orig=x;
    remove=1;
    x_sorted=sort(x);
    count=1;
    while isempty(remove)~=1
 15     temp=length(x_sorted);
```

```
          mean_x=mean(x_sorted);
          std_x=std(x_sorted);

20        T_min=(mean_x-x_sorted(1))/std_x;
          T_max=(x_sorted(temp)-mean_x)/std_x;

          DF=temp-2;
          x=(1-.95)/2;
25        T_crit=tinv((1-x),DF);
          G=(temp-1)/sqrt(temp)*sqrt(T_crit^2/(temp-2+T_crit^2));

          mag_min=[];
          mag_max=[];
30        if T_min>G
              mag_min=T_min-G;
          end
          if T_max>G
              mag_max=T_max-G;
35        end

   % Deside which point in sorted data to get rid of
          if isempty(mag_min)+isempty(mag_max)==2
              % do not remove any points
40            remove=[];
          elseif isempty(mag_min)==1
              % remove max value
              remove=temp;
          elseif isempty(mag_max)==1
45            % remove min value
              remove=1;
          else
              largest=max(mag_min,mag_max);
              if largest==mag_min
50                % remove min value
                  remove=1;
              else
                  % remove max value
                  remove=temp;
55            end
          end
          remove;
          if isempty(remove)==1
              break
60        end

          % figure out which point in real data was removed
          for i = 1:length(x_orig)
              if x_sorted(remove)==x_orig(i)
65                Removed(count)=i;
                  count=count+1;
                  break
              end
          end
70        x_sorted(remove)=[];
   end
```

Listing C.7   Water Inertia and Mass Iteration Code.

```
function [m_est,I_fuel]=Inertia_to_mass(I_baseline,I_test,L,rho)

syms r_sym V_sym

% rho=1000;          % kg/m^3
I_fuel=0;
change=1;

while change>0.00001
    I_fuel_old=I_fuel;
    m_est=(I_test-I_baseline-I_fuel)/L^2;

    V=(m_est/2)/rho;
    r=8*0.0254;

    g = solve('V_sym=pi*(2*r_sym^3+3*h*r_sym^2-h^3)/3','h');

    temp=real(subs(g,{r_sym,V_sym},{r,V}));

    for i = 1:length(temp)
        if temp(i)>r;
        elseif temp(i)<-r;
        else
            h=temp(i);
        end
    end
    I_fuel=2*(1/2*rho*pi*(8*r^5/15+h*r^4-2*h^3*r^2/3+h^5/5));
    change=abs(I_fuel_old-I_fuel);
end
```

Listing C.8   Data Progression Plotting Code.

```
function []=data_progression_plot(data,Removed,name)

Data_progression(1)={data};
for i = 1:length(Removed)
    temp=data;
    temp(Removed(1:i))=[];
    Data_progression(i+1)={temp};
    clear temp
end

figure
for i = 1:length(Data_progression)

    Data=cell2mat(Data_progression(i));
    for k = 1:length(Data)+1
        Ref(k)=k;
    end
    Ref=[0 Ref];
    Mean_data=mean(Data)*ones(1,length(Ref));
    STD2_data_neg=(mean(Data)-2*std(Data))*ones(1,length(Ref));
    STD2_data_pos=(mean(Data)+2*std(Data))*ones(1,length(Ref));
    STD1_data_neg=(mean(Data)-1*std(Data))*ones(1,length(Ref));
    STD1_data_pos=(mean(Data)+1*std(Data))*ones(1,length(Ref));
```

```matlab
25      subplot(1,length(Data_progression),i)
        line(Ref,Mean_data,'Color','g','LineWidth',2)
        line(Ref,STD2_data_neg,'Color','r','LineStyle','--','LineWidth',2)
        line(Ref,STD2_data_pos,'Color','r','LineStyle','--','LineWidth',2)
        line(Ref,STD1_data_neg,'Color','y','LineStyle','--','LineWidth',2)
30      line(Ref,STD1_data_pos,'Color','y','LineStyle','--','LineWidth',2)

        hold on
        plot(Data,'bs','MarkerFaceColor','b')
        hold off
35
        xlabel('Run Number')
        xlim([Ref(1) Ref(length(Ref))])
        clear Ref Mean_data STD_data_neg STD_data_pos
    %         hist(Data,3)
40      if i == 1
            title(name)
        elseif i < length(Data_progression)
            title(['Iteration ' num2str(i-1)])
        else
45          title('Final Data')
        end
    end
```
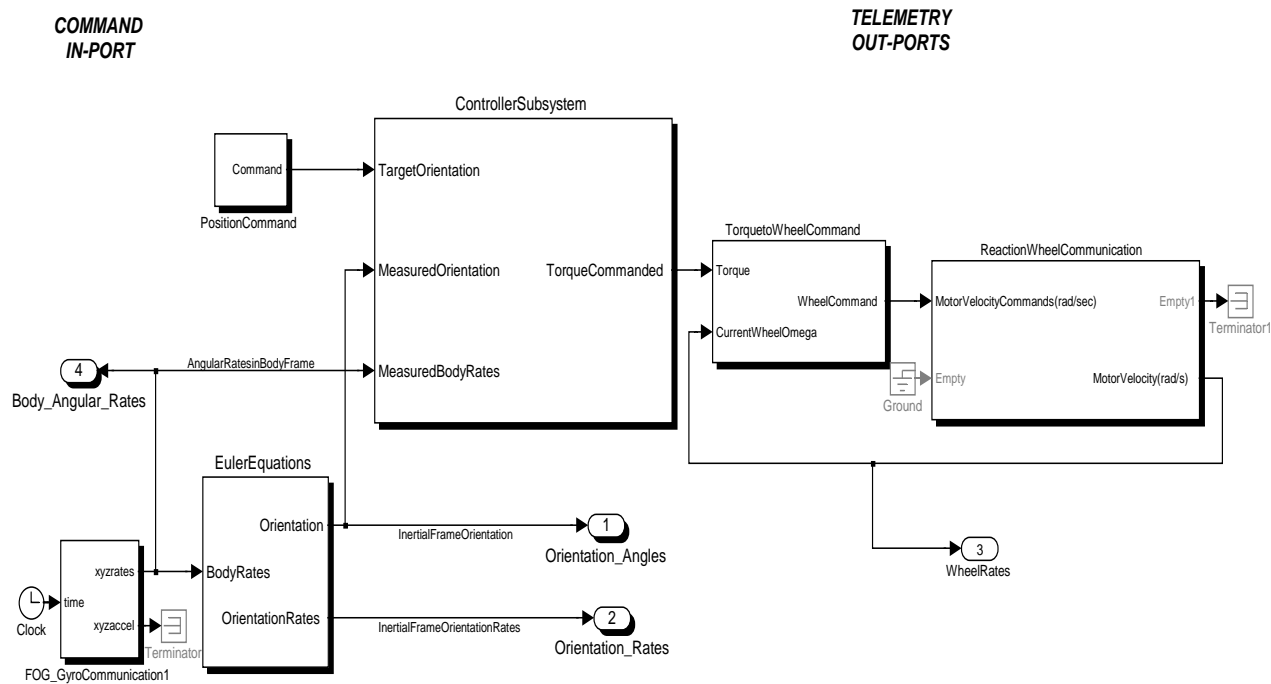
Figure D.1   SimSAT Main Diagram
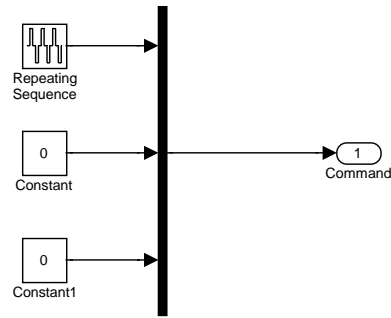
Figure D.2   Position Command for SimSAT
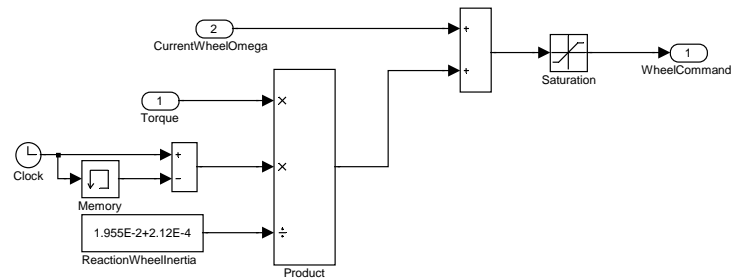
Figure D.3   SimSAT Control Input Converter Diagram
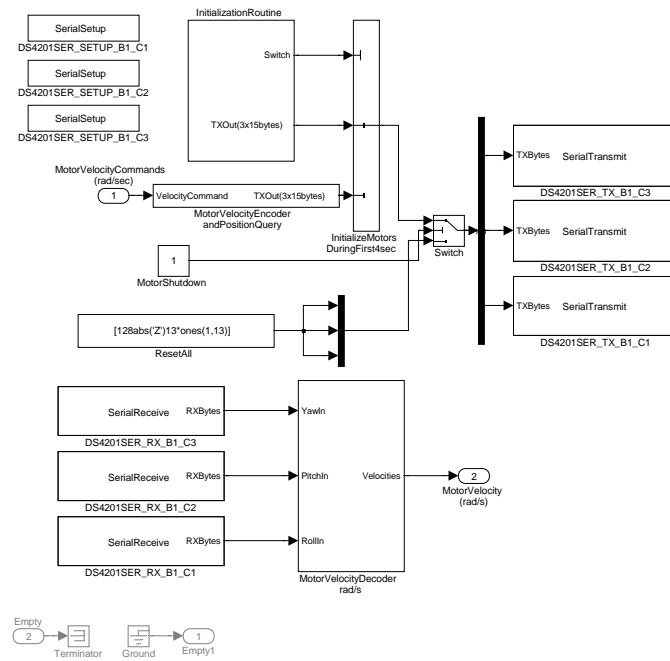
Figure D.4   Reaction Wheel Main Diagram
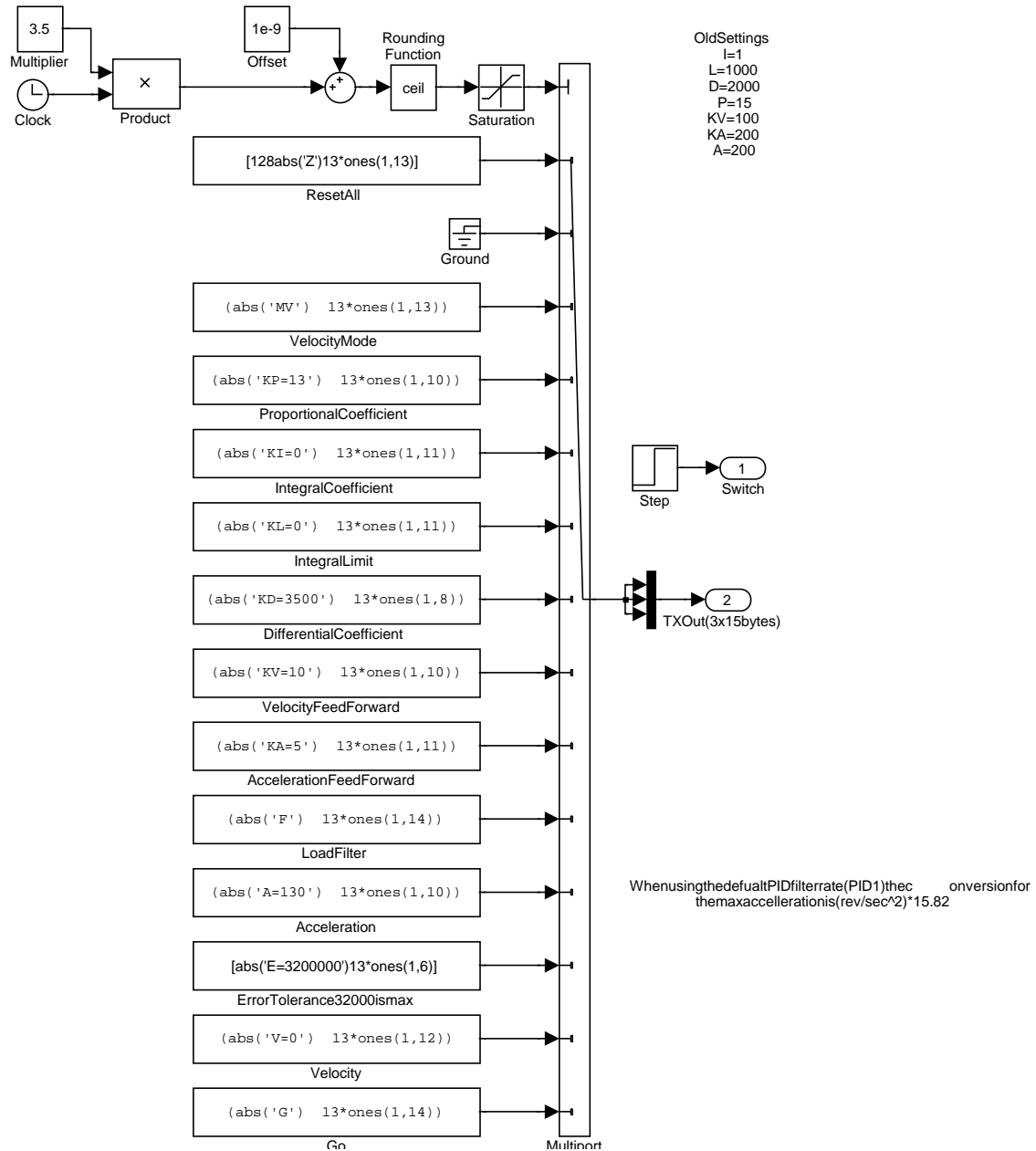
Figure D.5   Reaction Wheel Initialization
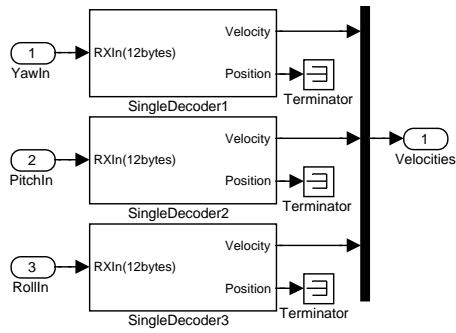
Figure D.6   Reaction Wheel Motor Velocity Decoder Top Level
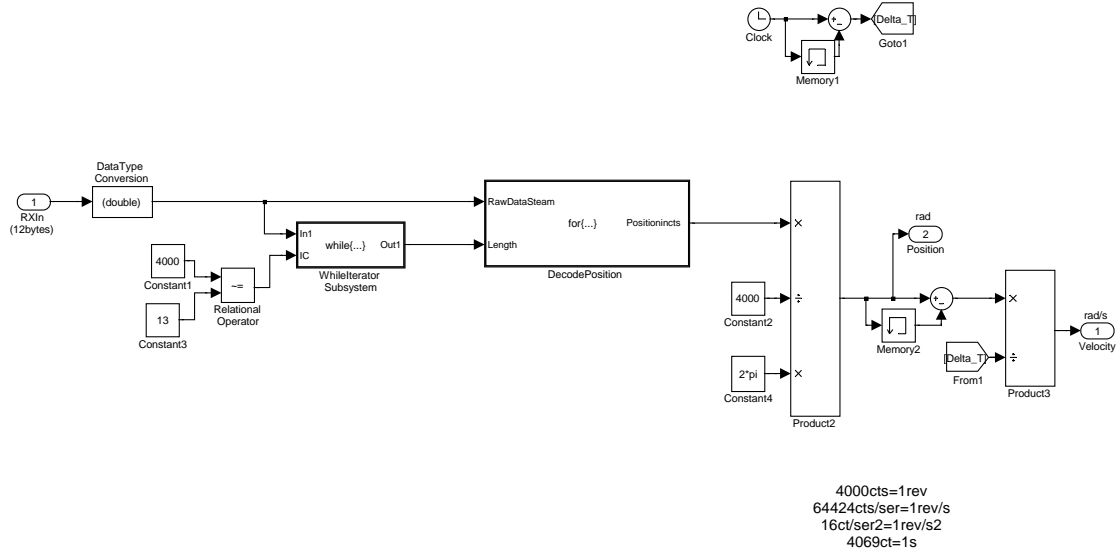
Figure D.7   Reaction Wheel Motor Velocity Decoder
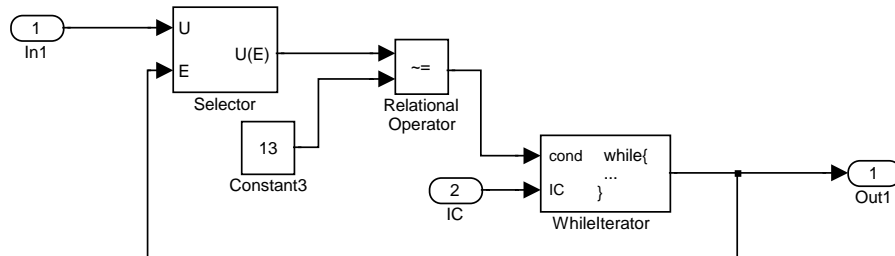
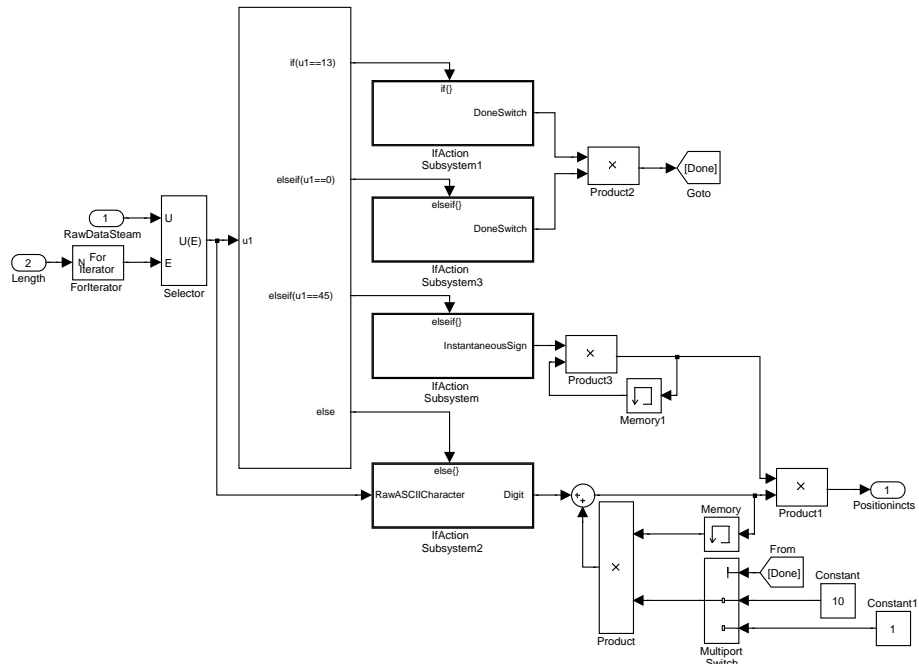Figure D.8   Reaction Wheel Position Filter
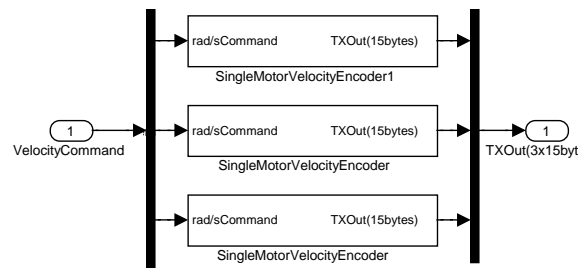
D-4

Figure D.9   Reaction Wheel Position Decoder

Figure D.10   Reaction Wheel Velocity Encoder and Position Query
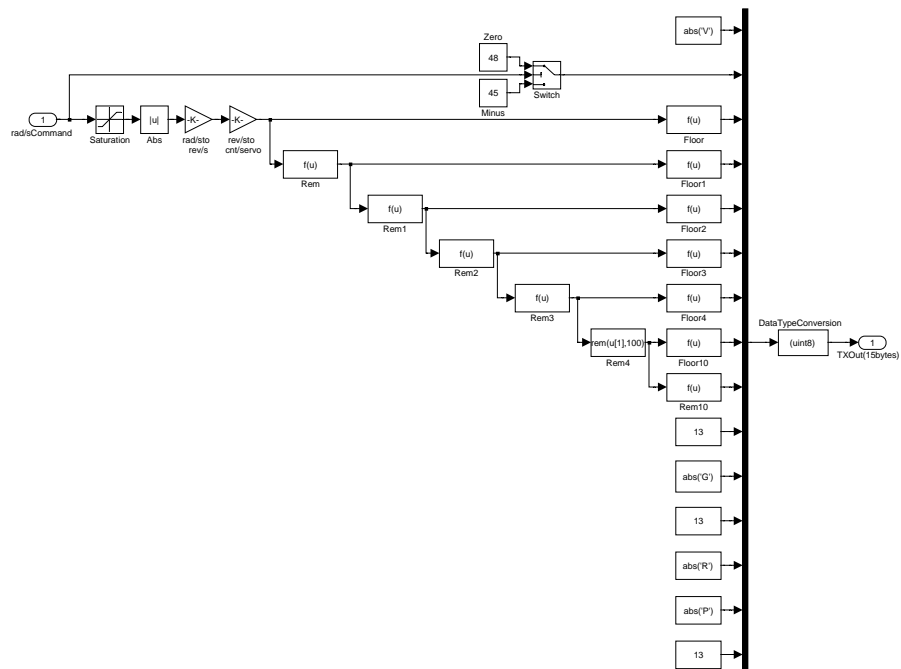
Figure D.11   Reaction Wheel Velocity Encoder
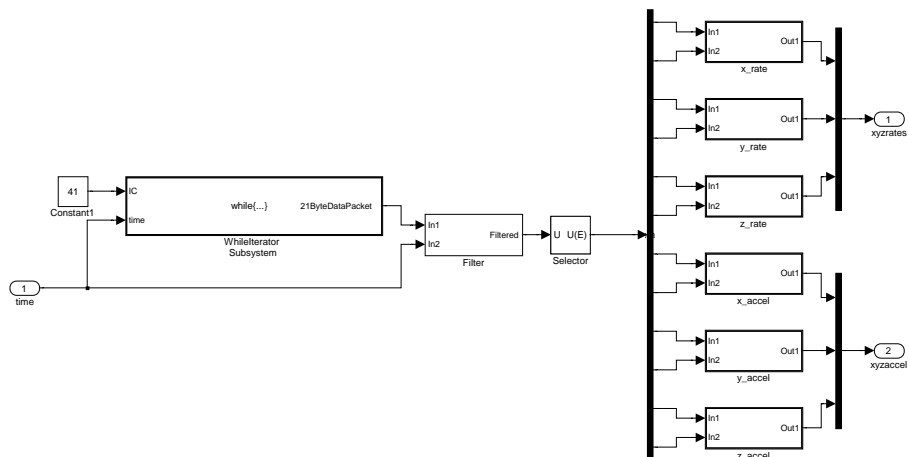
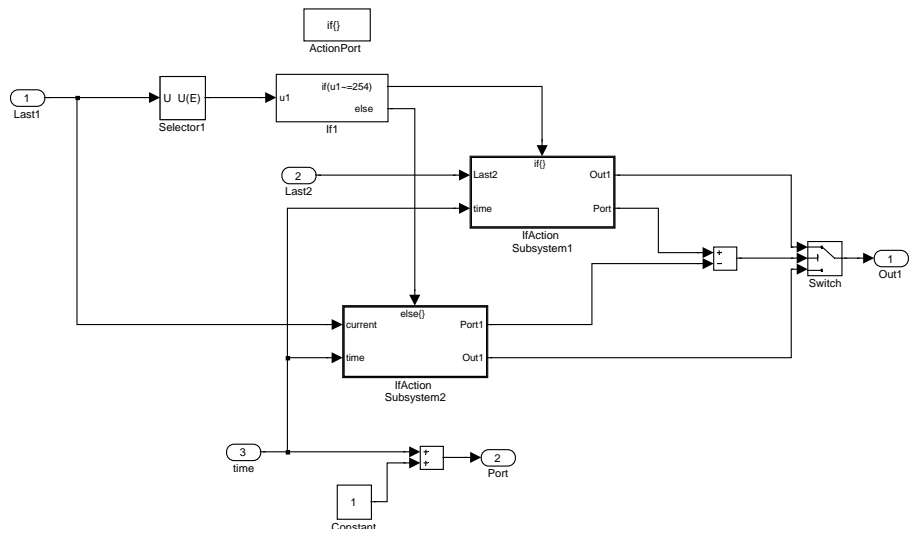Figure D.12   Gyro Main Diagram
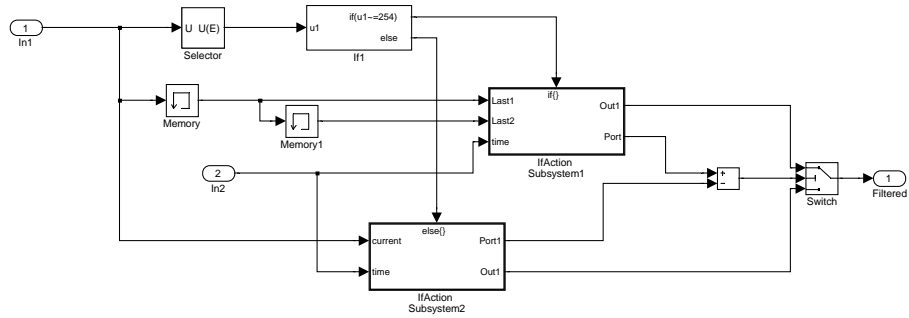
Figure D.13   Gyro Filter Level 1

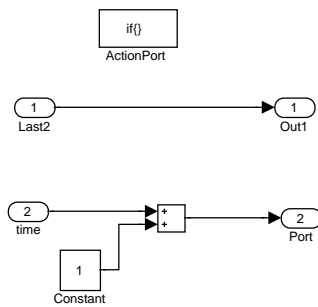Figure D.14   Gyro Filter Level 2

Figure D.15   Gyro Filter Level 3-1
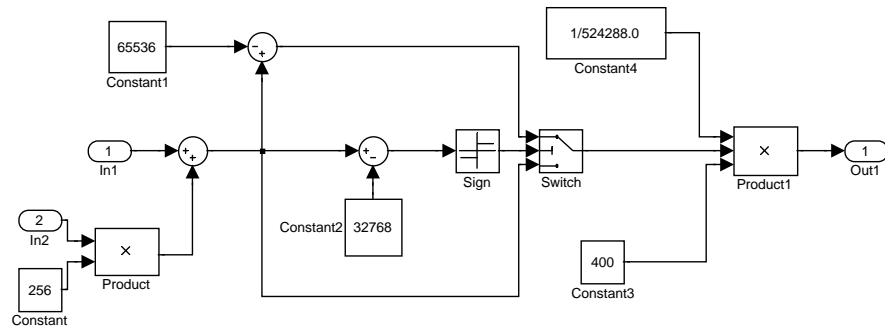
Figure D.16   Gyro Filter Level 3-2

Figure D.17   Gyro Filter Level 2

Figure D.18   Gyro While Iteration Level 1

Figure D.19   Gyro While Iteration Level 2 Rates

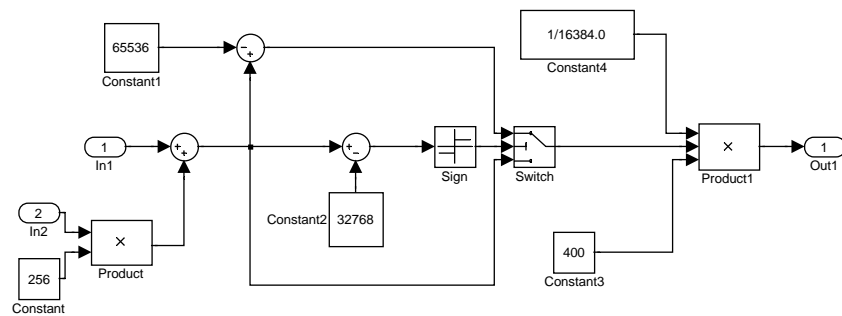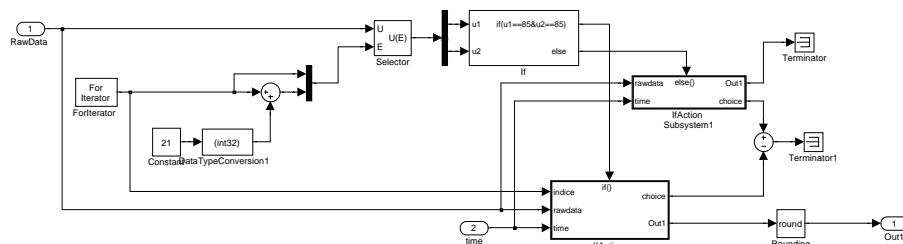Figure D.20   Gyro While Iteration Level 2 Accelerations

Figure D.21   Gyro While Iteration Level 2

Bibliography

1. "Matlab® R2006a: The Language of Technical Computing." The Mathwork, Inc.

2. "Grubbs Test." Engineering Statistics Handbook Homepage, http://www.itl.nist.gov/div898/handbook/index.htm, Accessed: 8 January 2007.

3. "http://hyperphysics.phy-astr.gsu.edu/hbase/isph.html." Accessed: 25 February 2007.

4. "http://www.robertniles.com/stats/stdev.shtml." Accessed: 28 January 2007.

5. "Interagency Report on Orbital Debris: Novemeber 1995." NASA Page on Orbital Debris, http://orbitaldebris.jsc.nasa.gov/faqs.html, Accessed: 30 January 2007.

6. "Second Report and Order: In the Matter of Mitigation of Orbital Debris." Federal Communication Commision Homepage, http://www.fcc.gov/, Accessed: 30 January 2007.

7. "Technical Report on Space Debris: United Nations 1999." NASA Page on Orbital Debris, http://orbitaldebris.jsc.nasa.gov/faqs.html, Accessed: 30 January 2007.

8. Adachi, Shuichi, et al. "On-orbit system identification experiements on Engineering Test Satellite-VI," *Control Engineering Practice*, *7*(7):831–841 (1999).

9. Brouard, L., et al. "Propellant Guaging Method Applied on ASTRIUM-SAS Telecommunication Satellite." *3rd International Conference on Spacecraft Propulsion*. 131–138. Cannes, France: European Space Agency, October 2000.

10. Colebank, James E., et al. *SIMSAT: A Satellite System Simulator and Experimental Test Bed for Air Force Research*. MS thesis, AFIT/GSE/GSO/ENY/99M-1, Air Force Institute of Technolgy, Wright Patterson AFB, March 1999.

11. Dabrowski, Vincent J. *Experimental Demonstration of an Algorithm to Detect the Presence of a Parasitic Satellite*. MS thesis, AFIT/GAE/ENY/03-2, Air Force Institute of Technolgy, Wright Patterson AFB, March 2003.

12. Devore, Jay and Roxy Peck. *Statistics: The Exploration and Analysis of Data*. Pacific Grove, CA: Duxbury, 2001.

13. Gangadharan, Sathya, et al. "Parameter Estimation of Spacecraft Fuel Slosh Model." *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Palm Springs, CA: American Institute of Aeronautics and Astronautics, April 2004.

14. Geitgey, Jason W. *The Determination of Remaining satellite Propellant Using Measured Moments of Inertia*. MS thesis, AFIT/GAE/ENY/06-J04, Air Force Institute of Technolgy, Wright Patterson AFB, June 2006.

15. "IADC Space Debris Mitigation Guideline: 15 October 2002." Inter-Agency Space Debris Coordination Committee Homepage, http://www.iadc-online.org, Accessed: 28 January 2007.

16. Kezirian, Michael T. "Fuel Slosh Dynamics in Spinning Spacecraft." *42nd AIAA/AS-ME/SAE/ASEE Joint Propulsion Conference and Exhibit*. Sacramento, CA: American Institute of Aeronautics and Astronautics, July 2006.

17. Matthijssen, R. and P. van Put. "State-of-the-Art Gauging Components for Improved Propellant Management on 3-Axis Stabilized Spacecraft." *42nd AIAA/AS-ME/SAE/ASEE Joint Propulsion Conference and Exhibit*. 2398–407. Sacramento, CA: American Institute of Aeronautics and Astronautics, July 2006.

18. Meirovitch, Leonard. *Methods of Analytical Dynamics*. Mineola, NY: Dover Publications, Inc., 1998.

19. Schlee, Keith, et al. "Advanced Method to Estimate Fuel Slosh Simulation Parameters." *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*. Tucson, AZ: American Institute of Aeronautics and Astronautics, July 2005.

20. Schlee, Keith, et al. "Parameter Estimation of Spacecraft Fuel Slosh." *47th AIAA/AS-ME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Newport, RI: American Institute of Aeronautics and Astronautics, May 2006.

21. Slotine, Jean-Jacques E. and Weiping Li. *Applied Nonlinear Control*. Upper Saddle River, New Jersey: Prentice Hall International Inc., 1991.

22. *Space Object Identification*. Contract N00014-97-D-2014/001, Albuquerque, NM: Schafer Corporation, February 1999.

23. Taylor, John K. and Cheryl Cihon. *Statistical Techniques for Data Analysis*. New York, NY: Chapman and Hall/CRC, 2004.

24. Viberg, M. "Subspace-based methods for the identification of linear time-invariant systems," *Automatica*, *31*(12):1835–51 (1995).

25. Wie, Bong. *Space Vehicle Dynamics and Control*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc. 1998.

Vita

Lieutenant Michael S. Hines was raised a military brat and graduated in Oklahoma from Deer Creek High School in 2001. He applied and was accepted into the United States Air Force Academy where he graduated with a Bachelor of Science degree in Astronautical Engineering in June 2005.

Following graduation he was accepted into the Air Force Institute of Technology to earn his Master's Degree in Astronautical Engineering. Upon graduation, he will be assigned to the Air Force Research Lab Detachment 9 in Kirtland AFB, New Mexico.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
22–03–2007	Master's Thesis	Sep 2006 – Mar 2007

4. TITLE AND SUBTITLE

FUEL ESTIMATION USING DYNAMIC RESPONSE

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hines, Michael S., Second Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GA/ENY/07-M12

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

APPROVAL FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

New regulations governing satellites in geostationary orbit require satellites to transfer into a parking orbit as part of the decommissioning process. These regulations increase the demand for accurate fuel estimation techniques for satellites. This study estimates the change in fuel mass from the dynamic response of the Air Force Institute of Technology's simulated satellite (SimSAT) to known control inputs. With an iterative process, the moment of inertia of SimSAT about the yaw axis was estimated by matching a model of SimSAT to the measured angular rates. A change in fuel mass was then estimated from the known relation between the change in moment of inertia to the change in fuel mass.
Fuel masses of 1, 2, and 3 kilograms were estimated. The fuel estimation process developed in this study was able to estimate the fuel as $1.5664 \pm 3.7157\ kg$, $2.8880 \pm 3.8875\ kg$, and $3.9114 \pm 3.4648\ kg$ respectively. While the theory behind the estimation process is sound, the implementation still requires work.

15. SUBJECT TERMS

Moment of Inertia Estimation; Fuel Estimation; System Identification; Satellite Simulator; Hardware Simulation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Nathan A. Titus, Lt Col, USAF (ENY)
U	U	U	UU	102	**19b. TELEPHONE NUMBER** *(include area code)* (937) 785–3636, ext 7469

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18